# How to Prolong the Lifetime of Wireless Sensor Networks

Giuseppe Anastasi*, Mario Di Francesco
*Department of Information Engineering*
*University of Pisa, Italy*

Marco Conti, Andrea Passarella
*Institute for Informatics and Telematics (IIT)*
*National Research Council, Italy*

**Contents**



* Corresponding author. Contact address: Giuseppe Anastasi, via Diotisalvi 2, 56122 Pisa (Italy). Phone: +39 050 2217 559. Fax: +39 050 2217 600. E-mail: giuseppe.anastasi@iet.unipi.it

This work was carried out under the financial support of the Italian Ministry for Education and Scientific Research (MIUR) in the framework of the PRIN project WiSeMaP (Wireless Sensor Networks for Monitoring Natural Phenomena, grant # 2005090483).

## I. INTRODUCTION

A wireless sensor network consists of a large number of sensor nodes deployed over a geographical area for monitoring physical phenomena like temperature, humidity, vibrations, seismic events, and so on. Each sensor node is a tiny device that includes three basic components: a sensing subsystem for data acquisition from the physical surrounding environment, a processing subsystem for local data processing and storage, and a wireless communication subsystem for data transmission to a central collection point (sink node or base station). In addition, a power source supplies the energy needed by the device to perform the programmed task. This power source often consists of a battery with a limited energy budget. In addition, it could be impossible or inconvenient to recharge the battery, because nodes may be deployed in a hostile or unpractical environment. On the other hand, the sensor network should have a lifetime long enough to fulfill the application requirements. In many cases a lifetime in the order of several months, or even years, may be required. Therefore, the crucial question is: "*how to prolong the network lifetime to such long time?*"

In some cases it is possible to scavenge energy from the external environment (e.g., by using solar cells as power source). However, external power supply sources often exhibit a non-continuous behavior so that an energy buffer (a battery) is needed as well. In any case, energy is a very critical resource and must be used very sparingly. Therefore, energy saving is a key issue in the design of systems based on wireless sensor networks.

Experimental measurements have shown that data transmission is very expensive in terms of energy consumption, while data processing consumes significantly less [1]. The energy cost of transmitting a single bit of information is approximately the same as that needed for processing a thousand operations in a typical sensor node [2]. The energy consumption of the sensing subsystem depends on the specific sensor type. In many cases it is negligible with respect to the energy consumed by the processing and, above all, the communication subsystems. In other cases, the energy expenditure for data sensing may be comparable to, or even greater than, the energy needed for data transmission.

The lifetime of a sensor network can be extended by jointly applying different techniques. Energy efficient protocols are aimed at minimizing the energy consumption during network activities. However, a large amount of energy is consumed by node components (CPU, radio, etc.) even if they are idle. Energy or power management schemes are thus used for switching off node components that are not temporarily needed. Finally, it's convenient to consider the energy consumption problem on a system basis rather than on a component/protocol basis. For this purpose, a cross-layer approach can be exploited to reduce the energy expenditure through the entire protocol stack.

Based on the above results several energy conservation schemes have been proposed. They are mainly aimed at minimizing the energy consumption of the communication subsystem. With regard to this, there are two main approaches to energy conservation: *in-network processing* and power saving through *duty cycling*. In-network processing consists in reducing the number of information to be transmitted by means of compression or aggregation techniques. It typically exploits the temporal or spatial correlation among data acquired by sensor nodes. On the other hand, duty cycling schemes define coordinated sleep/wakeup schedules among nodes in the network.

In this chapter we will survey the main techniques used for energy conservation in sensor networks. Specifically, we focus primarily on duty cycling schemes which represent the most suitable technique for energy saving. However, we will also survey the main energy-efficient networking protocols proposed for sensor networks (e.g., routing protocols, transport/congestion control protocols, and so on). Furthermore, we show that cross-layering is a must in the design of any system based on sensor networks. On the other hand, we will not consider in-network processing techniques as they are typically application-dependent.

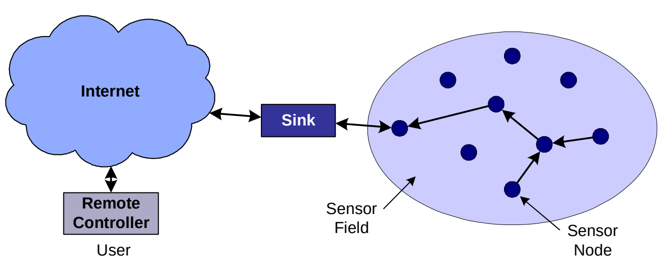


FIG. 1. Sensor network architecture.

In this chapter we will refer mainly to the sensor network model depicted in FIG. 1. and consisting of one (or more) sink(s) and a high number of sensor nodes deployed over a large geographic area (*sensing field).* Data are transferred from sensor nodes to the sink through a multi-hop communication paradigm [3]. Both the sink and the sensor nodes are assumed to be static (static sensor network). However, we will also briefly discuss energy conservation schemes for sensor networks with mobile elements (data mules).

The rest of the chapter is organized as follows. Section II surveys the main techniques for harvesting energy from the external physical environment. Section III discusses the general approaches to energy saving in sensor nodes, and introduces the taxonomy of energy conservation schemes. Section IV analyzes the main topology control protocols. Sections V and VI are devoted to power management schemes that can be implemented either as general protocols on top of a MAC protocol (Section V), or within the MAC protocol itself (Section VI). Section VII highlights the benefits in terms of energy saving of taking a cross-layer approach in the design of systems based on sensor networks. Energy harvesting, topology control, power management and cross-layering can be regarded as building blocks to design energy-efficient networking protocols which are surveyed in Section VIII.

## II. HARVESTING ENERGY FROM THE EXTERNAL ENVIRONMENT

The idea of scavenging energy from the external environment to feed electronic devices is not new. For example, electronic calculators powered by light sources have been sold since a long time ago. The new challenge is how to harvest enough energy to sustain the operation of devices. Investigating this direction is very important, for several reasons. Firstly, energy harvested from the environment is pollution free. Secondly, being renewable, it potentially allows devices to run unattended for virtually unlimited time.

Energy harvesting for sensor nodes (and more generally for portable computers) is still in its early stages, and is gaining momentum in the research community [4], [5]. A first research direction is collecting energy from electromagnetic fields. The most popular and developed example is getting energy from light sources via solar cells [6]. Unfortunately, current technology allows conversion efficiency just between 10% and 30%, thus requiring too large surfaces to produce reasonable amounts of energy [7]. Should conversion efficiency improve, in many cases this technology could replace batteries [8].

It is also possible to harvest energy from Radio Frequency (RF) signals. Actually, this is the way passive RF tags work. This approach can be extended to more complex devices, as well. For example, researchers are trying to feed sensor nodes through the RF signal sent by a reader. While the physical principle is exactly the same as in RF tags, the power required for feeding a sensor node is quite higher [9], making such a technique a challenging one.

Thermal gradients are another possible source of energy harvesting. The Carnot cycle is the physical principle behind this approach. For example, the Seiko Thermic wristwatch exploits the thermal gradient between the human body and the environment [7]. Also in this case, the conversion efficiency is the main problem, especially when the thermal gradient is small. This technique could be used for wearable sensor nodes, but it is unsuitable for sensor networks deployed in a sensing area.

Radioactivity has also been proposed as a source of energy for small devices [10]. The typical limited size of the radiating material avoids safety and health problems. This technology is particularly suitable for devices operating with very limited power (i.e., tens of μW) for very long time. Indeed, the limit in time of such a system is governed by the half-life of the radiating material, which can be in the order of hundreds of years [10].

Mechanical movements can be exploited to scavenge energy as well. For example, vibrations in the environment can be converted through piezoelectric materials. Research in this field is already quite developed, so that it has been possible to feed an off-the-shelf Mica2Dot Mote operating at a 1% duty cycle just by means of such a technique [11]. Human movements can be also used to collect energy. Self-winding wristwatches date back a long ago, as they have been diffused since 1930s. More recently (1997), the same principle has been used to build windup radios to be used when battery availability is an issue [8]. Finally, it has also been proposed to harvest energy by heel strikes when people walk. It has been proved that this approach can produce an average power in the order of 250-700 mW, thus representing a very promising direction [7].

Even though in the very long term energy harvesting techniques might represent the main power source for sensor nodes, in the meanwhile the conversion process is not efficient enough. Energy scavenging can thus be used just to power very simple devices (such as RFID), or as a complementary power source, e.g., to replenish a battery in the background. In general, the main issue seems not to be the amount of *energy* that can be collected through harvesting (which is virtually infinite), but the amount of *power*, which is quite limited [3]. Therefore, even when using systems to scavenge energy from the external environment, energetic resources at sensor nodes must be used judiciously. Hence, energy harvesting and energy conservation are two key principles around which sensor networks and systems should be designed. In the next sections we will survey the main techniques to reduce energy consumption in sensor networks, thus prolonging their lifetime.

## III. REDUCING ENERGY CONSUMPTION

### A. General Approaches To Energy Saving

Energy is a critical resource in wireless sensor networks, even when it is possible to harvest energy from the external environment. Therefore, the key question to answer when designing a sensor network based system is the following one. “*How to minimize the energy consumption of sensor nodes while meeting application requirements?*”.

To answer the above question it is important to know how much power each node component dissipates during normal operating conditions, i.e., which are the power dissipation characteristics of sensor noses [1].

FIG. 2 shows the architecture of a typical wireless sensor node. It consists of four main components: (i) a *sensing subsystem* including one or more sensors (with associated analog-to-digital converters) for data acquisition; (ii) a *processing subsystem* including a micro-controller and memory for local data processing; (ii) a *radio subsystem* for wireless data communication; and (iv) a *power supply unit*. Depending on the specific application, sensor nodes may also include additional components such as a *location finding system* to determine their position, a *mobilizer* to change their location or configuration (e.g., antenna’s orientation), and so on. However, as the latter components are optional, and only occasionally used, we will not take them into account in the following discussion.

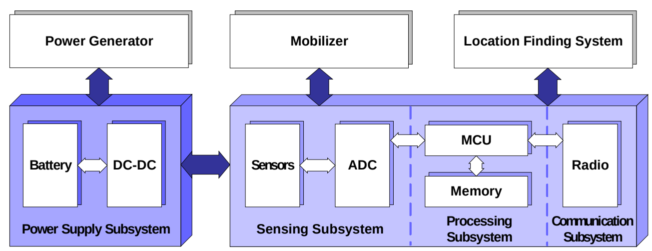


FIG. 2: Architecture of a typical wireless sensor node.

Obviously, the power breakdown heavily depends on the specific node. In [1] it is shown that the power characteristics of a Mote-class node are completely different from those of a Stargate node. However, the following remarks generally hold [1].

- The radio subsystem is the component that accounts for the largest energy consumption. A comparison of computation and communication costs has shown that transmitting one bit over a distance of 100 m consumes approximately the same energy as executing 3000 instructions [2]. Therefore, to reduce energy consumption the number of communications should be minimized, even at the cost of increasing data processing.
- Due to the small transmission distances, typically the power consumed for receiving may be greater than the power consumed for transmitting. Therefore, there is no real advantage in minimizing the number of transmissions. Instead, a power-efficient design should minimize the number of receptions.
- The power consumed when the radio is *idle* (i.e., it is neither receiving nor transmitting data) is approximately the same as in transmit/receive mode. Therefore, there is no real advantage in maintaining the radio in idle mode.

- The power consumption of the sensor node depends on the operational mode of the components. For example, putting the radio in the sleep mode reduces significantly the node power consumption. Therefore, node components, and specifically, the radio subsystem, should be put in sleep mode whenever possible.

Based on the above general remarks, several approaches can be exploited, even simultaneously, to reduce power consumption in wireless sensor networks. The most effective way is putting the radio transceiver in the (low-power) sleep mode whenever communication is not required. Ideally, the radio should be switched off as soon as there is no more data to send/receive, and should be resumed as soon as a new data packet becomes ready. This way nodes alternate between active and sleep periods depending on network activity. This behavior is usually referred to as *duty cycling*, and *duty cycle* is defined as the fraction of time nodes are active during their lifetime.

Obviously, from the power saving standpoint, the duty cycle should be as low as possible. However, as sensor nodes perform a cooperative task, they need to coordinate their sleep/wakeup times. A *sleep/wakeup scheduling algorithm* is required to this end. The sleep/wakeup scheduling algorithm is typically a distributed algorithm based on which sensor nodes decide when to transition from active to sleep, and back. It allows neighboring nodes to be active at the same time, thus making packet exchange feasible even when nodes operate with a low duty cycle (i.e., they sleep for most of the time)

Duty cycling reduces significantly the energy consumption of sensor nodes as, ideally, it keeps nodes active only when there is network activity. Actually, it is the most effective approach to energy conservation. However, additional energy savings can be achieved through an energy-efficient design of applications and networking protocols. The goal is to develop applications and networking protocols that perform their specific task by minimizing network activity.

At the application layer energy-efficiency can be achieved through *in-network processing* (also called *data aggregation*). In-network processing basically consists in reducing the amount of data to be transmitted to the sink node, even shifting some processing from the sink to intermediate nodes. For example, it is possible to aggregate packets or compress data by exploiting the spatial and/or temporal correlation in the acquired data. Furthermore, in many cases the application just requires aggregate information instead of raw data read by sensor nodes. For example, the sink node may be interested in knowing the maximum (or minimum) temperature within the sensing area. In such a case, there is no need to collect all temperature values at the sink node. The maximum (minimum) value can be computed on the fly by intermediate nodes in a cooperative way. When an intermediate node receives data from its neighbors, it extracts and forwards upstream only the maximum (minimum) value. Needless to say, the most appropriate in-network processing technique depends on the specific application and must be tailored to it. An interesting recent example of such technique is presented in [12]. In this paper, authors trade energy consumption for data quality: the higher the accuracy of the reported data, the higher the energy spent in the network. Such an approach leverages the evidence that often even rough data are sufficient for the sink to gather enough information from the environment.

Energy efficiency is also the key issue of any networking protocol for wireless sensor networks. Due to energy limitations, networking protocols must be designed to perform their specific task (e.g., routing) by minimizing energy consumption, possibly at the cost of decreased performance (e.g., energy saving is often traded off with latency or throughput). In addition, networking protocols must be aware of the sleep/wakeup algorithm used to implement duty cycling. In many cases the sleep/wakeup scheme is strictly tied with the networking protocol itself. For example, many MAC protocols for wireless sensor networks include a sleep/wakeup scheme for low duty cycle operations (see Section VI).

However, optimizing each single networking protocol is of limited help. It may also happen that reducing the energy consumption of a single protocol increases the energy consumption of the overall node [3]. What is really important is to minimize the energy consumption of the entire sensor node. To this end, a cross-layer design approach is much more appealing as it allows to face the energy problem from a system perspective.

In the next subsection we will introduce the taxonomy and the classification of duty cycling schemes. Then, we will survey the main proposals falling in the different categories (Section IV through Section VI). Finally, we will shed some light to cross-layer design (Section VII), and will survey the main networking protocols for wireless sensor networks tailored to reducing energy consumption.

### B. Taxonomy of Duty Cycling Schemes

As shown in FIG. 3, duty cycling can be achieved through two different and complementary approaches. From one side it is possible to exploit node redundancy, which is typical in sensor networks, and adaptively select only a minimum subset of nodes to remain active for maintaining connectivity. Nodes that are not currently needed for ensuring connectivity can go to sleep and save energy. Finding the optimal subset of nodes that guarantee connectivity is referred to as *topology control*. Therefore, the basic idea behind topology control is to exploit the network redundancy to increase the network longevity. On the other hand, active nodes (i.e., nodes selected by the topology control protocol) do not need to maintain their radio continuously on. They can switch off the radio (i.e., put it in the low-power sleep mode) when there is no network activity, thus alternating between sleep and wakeup periods. Throughout we will refer to duty cycling operated on active nodes as *power management*. Therefore, topology control and power management are complementary techniques that implement duty cycling with different granularity.

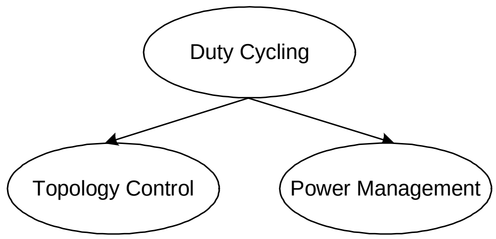


FIG. 3: Taxonomy of duty cycling schemes.

In the following two subsections we will provide a finer classification of topology control and power management technique, respectively.

### 1. *Topology Control*

The concept of topology control is strictly associated with that of network redundancy. Dense sensor networks typically have some degree of redundancy. In many cases network deployment is done at random, e.g., by dropping a large number of sensor nodes from an airplane. Therefore, it may be convenient to deploy a number of nodes greater than necessary to cope with possible node failures occurring during or after the deployment. In many contexts it is much easier to deploy initially a greater number of nodes than re-deploying additional nodes when needed. For the same reason, a redundant deployment may be convenient even when nodes are placed by hand [13].

If the number of nodes is redundant, it follows that not all nodes are needed for normal activities required by the application(s). Therefore, a fraction of them may be kept inactive. Keeping redundant nodes inactive also helps in avoiding interferences between neighboring nodes. Inactive nodes will be switched on when necessary (for example, when a node fails or runs out of energy). Topology control protocols are thus aimed at dynamically adapting the network topology, based on the application needs, so as to allow network operations while minimizing the number of active nodes (and, hence, prolonging the network lifetime).

Before proceeding on it may be worthwhile to point out that the term “topology control” has been used with a larger scope than that defined above. Some authors include in topology control also techniques that are aimed at super-imposing a hierarchy on the network organization (e.g., clustering techniques) to reduce energy consumption. In addition, the terms “topology control” and “power control” are often confused. However, power control refers to techniques that adapt the transmission power level to optimize a single wireless transmission. Even if the above techniques are related with topology control, in accordance with [14], we believe that they cannot be classified as topology control techniques. Therefore, in the following we will refer to topology control as a mean to reduce energy consumption by exploiting node redundancy.

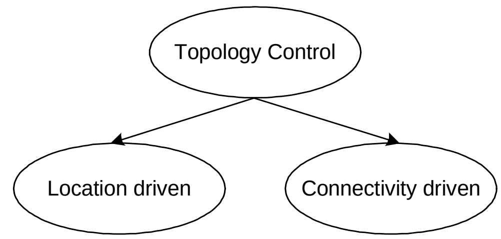


FIG. 4: Classification of topology control protocols.

There are two main issues that a topology control protocol must address:

(i) how many sensor nodes to activate?

(ii) which nodes to turn on, and when?

As far as point (i), it is worthwhile to highlight that, if there are too few active nodes, the distance between neighboring nodes is large and the energy required to transmit a packet becomes relevant. In addition, packet loss increases. On the other hand, if there are too many active nodes, not only they use unnecessary energy, but they may also interfere with each other.

Several criterions can be used to decide which nodes to activate/deactivate, and when. From this regard, topology control protocols can be broadly classified in the following two categories:

- *Location driven*. The decision about which node to turn on, and when, is based on the location of sensor nodes which is assumed to be known [15].
- *Connectivity driven*. Sensor nodes are dynamically activated/deactivated in such way to ensure network connectivity [16], [17], or complete sensing coverage [18].

Topology control protocols can extend the network longevity by a factor of 2-3 (depending on the network redundancy) with respect to a network with nodes always on [13], [19]. However, many sensor network applications require a much longer network lifetime, e.g., 100 times longer [19]. To further increase network longevity topology control must be combined with power management which introduces duty cycling even in active (i.e., non-redundant) nodes [20].

### 2. *Power Management*

Power management techniques can be subdivided into two broad categories depending on the layer of the network architecture they are implemented at. As shown in FIG. 5, power management protocols can be implemented either as independent sleep/wakeup protocols running on top of a MAC protocol (typically at the network or application layer), or strictly integrated with the MAC protocol itself. The latter approach permits to optimize medium access functions based on the specific sleep/wakeup pattern used for power management. On the other hand, independent sleep/wakeup protocols permit a greater flexibility as they can be tailored to the application needs, and can be used with *any* MAC protocol.

Independent sleep/wakeup protocols can be classified in three broad categories, depending on the general approach they take to decide when sensor nodes should be switched on: on-demand, scheduled rendezvous, and asynchronous protocols (see FIG. 5). It may be worthwhile to recall here that sensor nodes must coordinate their wakeup periods in order to make multi-hop communication feasible and, hopefully, efficient.

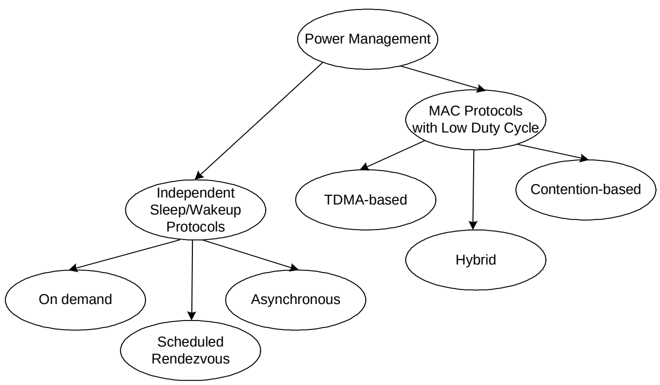


FIG. 5: Classification of power management techniques.

*On-demand* protocols [21], [22], [20] take the most intuitive approach to power management. The basic idea is that a node should wakeup only when another node wants to communicate with it. This maximizes energy saving since a node remains active only for the minimum time required for communication. In addition, there is only a very limited impact on latency because the corresponding node wakes up immediately as soon as it realizes that there is a pending message.

The main problem associated with on-demand schemes is how to inform the sleeping node that some other node is willing to communicate with it. Typically, such schemes use two different radio channels. The first channel is used for normal packet exchange (*data radio*), while the second one is used to awake a node when there is message ready for it (*wakeup radio*). The data radio is normally off, and is switched on only when a signal is received through the wakeup radio. Clearly, the wakeup radio should have a limited impact on the node's consumption. Different on-demand schemes differ in the way they use the wakeup radio. In many cases the power consumption of the wakeup radio is not very different from that of the data radio. Duty cycling scheme is thus used on the wakeup radio as well [22], [20]. Other works assume that the wakeup radio is very low-power and can thus be always on [23], [24], [25], [26]. The drawback is that the low-power wakeup radio typically has a communication range smaller than the data radio. This is a strong limitation since two neighboring nodes may be within each other's data radio transmission range but not within the wakeup radio range.

When a second (wakeup) radio is not available or convenient, an alternative is using a *scheduled rendezvous* approach [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39]. The basic idea behind scheduled rendezvous schemes is that each node should wakeup at the same time as its neighbors. Typically, nodes wake up according to a wakeup schedule, and remain active for a short time interval to communicate with their neighbors. Then, they go to sleep until the next rendezvous time. Different schemes differ in the sleep/wakeup pattern followed by nodes (see Section V-B). A drawback of the scheduled rendezvous schemes is that energy saving is obtained at the expense of an increased latency experienced by messages to travel through several hops. An additional drawback is that nodes must be synchronized.

In the literature several clock synchronization protocols (e.g., [40], [41]) have been proposed to keep nodes synchronized. However, maintaining a tight synchronization among nodes requires a high overhead in terms of exchanged control messages. This, of course, results in energy consumption. The basic assumption behind scheduled rendezvous schemes is that the energy spent for keeping nodes synchronized is largely compensated by the energy saving achieved through power management.

To avoid node synchronization we can use an *asynchronous* sleep/wakeup protocol [42], [43], [44]. In the asynchronous protocols a node can wakeup when it wants and still be able to communicate with their neighbors. This goal can be achieved by designing a sleep/wakeup scheme such that any two neighboring nodes always have overlapped active periods within a specified number of cycles. Asynchronous schemes are generally easier to implement and can ensure network connectivity even in highly dynamic scenarios where synchronous schemes (i.e., scheduled rendezvous) become inadequate. This greater flexibility is compensated by a lower energy efficiency. In the asynchronous schemes nodes need to wakeup more frequently than in scheduled rendezvous protocols. Therefore, asynchronous protocols usually result in a higher duty cycle for network nodes than their synchronous counterparts. In other words, they trade energy consumption for ease of implementation and robustness of network connectivity.

As shown in FIG. 5, MAC protocols with low duty cycle can be broadly subdivided into three main categories: TDMA-based, contention-based, and hybrid protocols.

TDMA (*Time Division Multiple Access*) schemes [45], [46], [47] naturally enable a duty cycle on sensor nodes as channel access is done on a slot-by-slot basis. Time is slotted and slots are arranged in frames. Within each frame slots are assigned to individual nodes and can be used for transmitting/receiving packets to/from other nodes. Nodes need to turn on their radio only during their own slots and can sleep during slots assigned to other nodes. In principle, this allows to limit the energy consumption to the minimum required for transmitting/receiving data. In practice, TDMA-based protocols have several drawbacks that compensate the benefits in terms of energy saving [48]. They lack flexibility, have limited scalability, and require tight synchronization among network nodes. In addition, it is hard to find a slot assignment which avoids interferences between neighboring nodes because the interference range is larger than the transmission range and, above all, it is time-varying [49]. Moreover, TDMA-based protocols perform worse than contention-based protocols in low traffic conditions. For all the above reasons they are not frequently used as stand-alone protocols.

*Contention-based* protocols [50], [51], [33], [43], [52] are the most popular class of MAC protocols for wireless sensor networks. They achieve duty cycling by tightly integrating channel access functionalities with a sleep/wakeup scheme similar to those described above. The only difference is that in this case the sleep/wakeup algorithm is not a protocol independent of the MAC protocol, but is tightly coupled with it.

Finally, *hybrid* protocols [53], [48] try to combine the strengths of TDMA-based and contention-based MAC protocols while offsetting their weaknesses. The intuition behind hybrid protocols is to adapt the protocol behavior to the level of contention in the network. They behave as a contention-based protocol when the level of contention is low, and switch to a TDMA scheme when the level of contention is high.

## IV. TOPOLOGY CONTROL PROTOCOLS

Wireless sensor networks typically have some degree of node redundancy due to several reasons: (**i**) nodes are often deployed at random; (**ii**) a number of nodes greater than necessary is usually deployed to cope with possible node failures during or after the deployment; (**iii**) it is often easier to initially deploy a greater number of nodes than re-deploying additional nodes when needed. Topology control protocols are aimed at exploiting such redundancy to prolong the network lifetime by activating only a minimum subset of nodes that ensure network connectivity. A detailed survey on topology control in wireless ad hoc and sensor networks is available in [14]. In this section we only review the main proposals for topology control in wireless sensor networks. According to the taxonomy introduced in Section III-B, topology control protocols can be distinguished in location-driven and connectivity-driven protocols.

GAF [15] (Geographical Adaptive Fidelity) is a location-driven protocol that reduces energy consumption while keeping a constant level of routing fidelity. It relies upon node location information that can be provided by a GPS (Global Positioning System) or some other location system. The sensing area where nodes are distributed is divided into small *virtual grids*. Each virtual grid is defined such that, for any two adjacent grids A and B, all nodes in A are able to communicate with nodes in B, and vice-versa (see Figure FIG. 6). All nodes within the same virtual grid are equivalent for routing, and just one node at time need to be active. Therefore, nodes have to coordinate each other to decide who can sleep and how long.

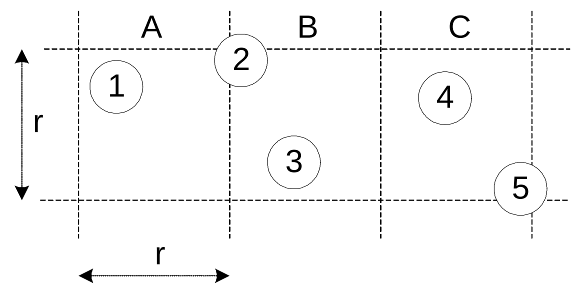


FIG. 6. Virtual grids in GAF.

In GAF nodes can be in one of the following states: *sleeping*, *discovery*, and *active* (see FIG. 7). Initially a node starts in the discovery state where it exchanges discovery messages with other nodes. Specifically, as soon as a node enters the discovery state, it sets a timer $T_d$. When the timer fires, the node broadcasts its discovery message and enters the active state. In the active state, the node sets up a timer $T_a$ to define how long it can stay active. While active, it periodically re-broadcasts its discovery message at intervals $T_d$. A node in the discovery or active state can change its state to sleeping when it detects that some other equivalent node will handle routing. Nodes in the sleeping state wake up after a sleeping time $T_s$, and go back to the discovery state.

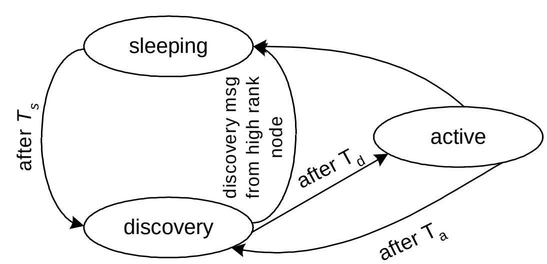


FIG. 7: State transitions in GAF.

In GAF load balancing is achieved through a periodic re-election of the leader (i.e., the node that will remain active to manage routing). The leader election is done by means of a rank-based election algorithm. The node with the highest rank becomes the node that will (temporarily) manage routing in the virtual grid. Node ranks are assigned in such way to maximize the network lifetime and are determined by several rules. First, a node in the active state has a higher rank than a node in the discovery state. This allows to quickly reach a condition where there is a single active node in each virtual grid. Second, for nodes that are in the same state, the node with the higher expected lifetime has the higher rank (possible ties are broken by considering node identifiers). To make energy consumption as uniform as possible, GAF uses the following strategy. After a node remains in the active state for a period $T_a$ it changes its state to discovery to allow other nodes to become active. As nodes in active state consume more energy than others, it is very likely that a node that was recently active has an expected lifetime lower than its neighbors in the virtual cell. Therefore, when it enters the discovery state and a new election procedure starts, it has less chances to be elected again.

GAF is independent from the routing protocol. It can be used with any existing routing protocol, and performs at least as well as normal routing protocols in terms of packet loss and message latency. On the other hand, it is able to conserve energy by exploiting node redundancy, thus allowing the network lifetime to increase in proportion to node density [15]. All nodes within a virtual grid are interchangeable from a routing perspective. This may result in an underutilization of radio coverage areas as nodes are forced to cover less than half the distance allowed by the radio range. In addition, GAF requires to know the exact location of each node in the network, which might be expensive to achieve. This drawback is overcome by *connectivity-driven* protocols. In such protocols nodes are able to discover and react to changes in the network topology, and decide whether to sleep or join the backbone based on connectivity information.

Span [17] is a connectivity-driven protocol that adaptively elects “coordinators” of all nodes in the network. Coordinators stay awake continuously and perform multi-hop routing, while the other nodes stay in sleeping mode and periodically check if there is a need to wake up and become a coordinator. The protocol achieves the following four goals. First, it ensures that there is always a sufficient number of coordinators so that every node is in the transmission range of at least one coordinator. Second, to spread energy consumption as uniformly as possible among network nodes Span rotates the coordinators. Third, it tries to minimize the number of coordinators (to increase the network lifetime) while avoiding a performance degradation in terms of network capacity and message latency. Fourth, it elects coordinators in a decentralized way by using only local information.

To guarantee a sufficient number of coordinators Span uses the following *coordinator eligibility rule*: if two neighbors of a non-coordinator node cannot reach each other, either directly or via one or more coordinators, that node should become a coordinator. However, it may happen that several nodes discover the lack of a coordinator at the same time and, thus, they all decide to become a coordinator. To avoid such cases nodes that decide to become a coordinator defer their announcement by a random *backoff delay*. If at the end of the backoff delay, the node has not yet received any announcement from other potential coordinators, it send its announcement and becomes a coordinator. Otherwise, it re-evaluates its eligibility based on announcement messages received, and makes its announcement if and only if the eligibility rule is still satisfied.

A key point in the above coordinator election algorithm is how to select the random backoff delay. Each node uses a function that generates random time by taking into account both the number of neighbors that can be connected by a potential coordinator node, and its residual energy. The fundamental ideas are that (**i**) nodes with a higher expected lifetime should be more likely to volunteer to become a coordinator; and (**ii**) coordinators should be selected in such a way to minimize their number. The node expected lifetime can be measured by the ratio $E_r/E_m$, where $E_r$ denotes the amount of residual energy, while $E_m$ gives the maximum amount of available energy ($E_r/E_m$ is thus the fraction of energy still available at the node). As far as point (**ii**) above, the *utility* of a node to become a coordinator is defined as follows. Let $N_i$ be the number of neighbors of node *i*, and let $C_i$ the number of additional pairs of nodes among these neighbors that would be connected if *i* decided to become a coordinator. Clearly, $0 \le C_i \le \binom{N_i}{2}$, and the utility of node *i* can be defined as $\frac{C_i}{\binom{N_i}{2}}$. If nodes with a large utility value become coordinators, a lower number of coordinators is required in total to guarantee that each node is in the transmission range of at least one coordinator. Therefore, nodes with a higher utility value should volunteer more quickly than those with smaller values. Based on the above remarks the following heuristic is used in [15] to derive the random backoff interval

$$backoff\_delay = \left( \left(1 - \frac{E_r}{E_m}\right) + \left(1 - \frac{C_i}{\binom{N_i}{2}}\right) + R \right) \cdot N_i \cdot T \quad (1)$$

where $R$ is a random value uniformly distributed in [0,1], and $T$ is round trip delay experienced by a small packet over the wireless link.

Each coordinator periodically checks if it can stop being a coordinator. A node should withdraw as a coordinator if every pair of its neighbors can communicate directly, or through some other coordinators. To avoid loss of connectivity in the time interval between the withdrawal message by a coordinator and the subsequent announcement by a new coordinator, the old coordinator continues its service for a short time after announcing its withdrawal. This allows the routing protocol to rely upon the old coordinator until the new one is available.

The Span election algorithm requires to know neighbor and connectivity information to decide whether a node should become a coordinator or not. Such information are provided by the routing protocol. Therefore, SPAN depends on the routing protocol and requires modification in the routing lookup process.

ASCENT [16] (Adaptive Self-Configuring sEnsor Networks Topologies) is another connectivity-driven protocol that, unlike Span, does not depend on the routing protocol and does not require to modify the routing state. In ASCENT a node decides whether to join the network or continue to sleep based on information about connectivity and packet loss that are *measured locally* by the node itself.

The basic idea of ASCENT is that initially only some nodes are *active*, while all other ones are *passive*, i.e., they listen to packets but do not transmit. If the number of intermediate nodes is not large enough, the sink node may experience a large message loss from sources. The sink then starts sending *help* messages to solicit neighboring nodes that are in the passive state (*passive neighbors*) to join the network by changing their state from passive to active (*active neighbors*). As soon as a node joins the network it signals the presence of a new active node by sending a *neighbor announcement message*. This process continues until the number of active nodes is such that the message loss experienced by the sink is below a pre-defined application-dependent threshold. The process will re-start when some future network event (e.g. a node failure) or a change in the environmental conditions causes an increase in the message loss.

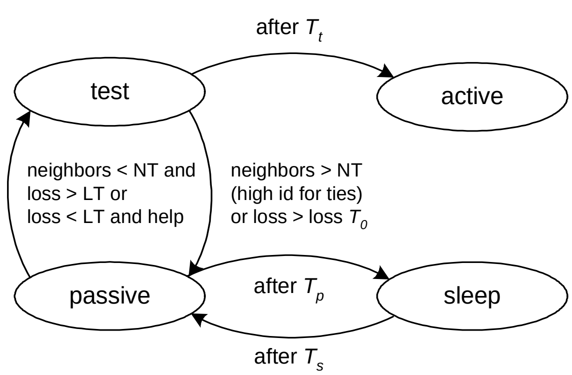


FIG. 8. State transitions in ASCENT.

The ASCENT protocol is slightly more complex than GAF and Span. The state transition diagram is shown in FIG. 8. Nodes may be in one of the following states: *sleep, passive, test, and active*. Initially nodes are in the test state. The rationale behind the test state is to check whether the addition of a new active node may help in improving network connectivity, while in the test state nodes exchange data and routing control messages. In addition, as soon as a node enters the test state, it sets a timer $T_t$, starts sending active neighbor announcements and, at the same time, monitors the network conditions. When the timer expires the node passes to the active state. However, the node transits to the passive state if one of the following two events is detected before the timer expiration:

(i) the number of active neighbors is above the *Neighbor Threshold* (NT);

(ii) the *Data Loss Rate* (Loss) is higher than that before entering the test state.

Due to (i), the number of active neighboring nodes cannot be larger than NT.

When a node enters the passive state it sets up a timer $T_p$. When $T_p$ expires the node enters the sleep state. However, if one of the following events occurs before the expiration of $T_p$ the node transits to the test state:

(i) the number of active neighbors is *below* the *Neighbor Threshold* (NT) ***and*** the *Data Loss Rate* (Loss) is *greater* than a predefined *Loss Threshold* (LT)

(ii) the *Loss Rate* (Loss) is *lower* than *Loss Threshold* and the nodes receive an *help* message.

In the passive state nodes have their radio on and listen to all packets transmitted by their active neighbors. However, they do not cooperate in forwarding data packets or exchanging routing control information. In other words, in the passive state nodes collect information about the network status without interfering with other nodes.

A node entering the sleep state sets up a timer $T_s$ and goes to sleep. When $T_s$ expires the node changes its state into passive. Finally, nodes in the active state forward data and routing control messages until they run out of energy. In the meanwhile, if the *Data Loss Rate* increases beyond the *Loss Threshold,* the active node sends *help* messages.

As mentioned above, ASCENT is independent of the routing protocol. In addition, it limits the packets loss due to collisions because the nodes density is regulated by the *Neighbor Threshold* value. Finally, the protocol has good scalability properties. On the other side, energy saving does not increase proportionally with the node density because it depends on passive-sleep cycle and not on the number of active nodes.

## V. GENERAL SLEEP/WAKEUP PROTOCOLS

In this section we will survey the main sleep/wakeup schemes implemented as independent protocols on top of the MAC protocol. According to the classification introduced in Section III-B, we will discuss on-demand, scheduled rendezvous, and asynchronous schemes, in separate subsections below.

### A. On-demand Schemes

On-demand schemes are based on the idea that a node should be awaken just when it has to receive a packet from a neighboring node. This minimizes the energy consumption and, thus, makes on-demand schemes particularly suitable for sensor network applications with a very low duty cycle (e.g., fire detection, surveillance of machine failures and, more generally, all event-driven scenarios). In such scenarios sensor nodes are in the *monitoring state* (i.e., they only sense the environment) for most of the time. As soon as an event is detected, nodes transit to the *transfer state*. On-demand sleep/wakeup schemes are aimed at reducing energy consumption in the monitoring state while ensuring a limited latency for transitioning in the transfer state.

The implementation of such schemes typically requires two different channels: a data channel for normal data communication, and a wakeup channel for awaking nodes when needed. Although it would be possible to use a single radio with two different channels, all the proposals rely on two different radios. This allows not to defer the transmission of signal on the wakeup channel if a packet transmission is in progress on the other channel, thus reducing the wakeup latency. The drawback is the additional cost for the second radio. However, this additional cost is limited as the radio system typically accounts for a small percent of the entire cost of a sensor node (less than 15% for a MICA mote [22]).

STEM (*Sparse Topology and Energy Management*) [22] uses two different radios for wakeup signals and data packet transmissions, respectively. The wakeup radio is not a low power radio (to avoid problems associated with different transmission ranges). Therefore, an asynchronous duty cycle scheme is used on the wakeup radio as well. Each node periodically turns on its wakeup radio for $T_{active}$ every $T$ duration. When a source node (*initiator*) has to communicate with a neighboring node (*target*), it sends a stream of periodic beacons on the wakeup channel. As soon as the target node receives a beacon it sends back a wakeup acknowledgement, and turns on its data radio. If a collision occurs on the wakeup channel, any node that senses the collision activates its data radio up (no wakeup acknowledgement is sent in case of collision). The wakeup beacon transmission is repeated up to a maximum time unless a wakeup acknowledgement is received from the target node.

In addition to the above beacon-based approach, referred to as STEM-B, in [54] the authors propose a variant (referred to as STEM-T) that uses a wakeup tone instead of a beacon. The main difference is that in STEM-T all nodes in the neighborhood of the initiator are awakened.

Both STEM-B and STEM-T can be used in combination with topology control protocols. For example, in a practical case the combination of GAF and STEM can reduce the energy consumption to about 1% of that of a sensor network with neither topology control nor power management. This increases the network lifetime of a factor 100 [54]. However, STEM trades energy saving for path setup latency. In STEM the inter-beacon period is such that there is enough time to send the wakeup beacon and receive the related acknowledgement. Let $T_{wakeup}$ and $T_{wack}$ denote the time required to transmit a wakeup beacon and the related acknowledgement, respectively. Since nodes are not synchronized, the receiver must listen on the wakeup radio for a time $T_{active}$ at least equal to $2T_{wakeup}+T_{wack}$ to ensure the correct reception of the beacon, i.e., $T_{active} \geq 2T_{wakeup}+T_{wack}$ (see also Section V-C). Clearly $T_{active}$ depends on the bit rate of network nodes. In low bit-rate networks the time between successive active periods ($T$) must be very large to allow a low duty cycle on the wakeup channel. This results in a large wakeup latency, especially in multi-hop networks with a large hop-count.

To achieve a tradeoff between energy saving and wakeup latency, [20] proposes a *Pipelined Tone Wakeup* (PTW) scheme. Like STEM, PTW relies on two different channels for transmitting wakeup signals and packet data, and uses a wakeup tone to awake neighboring nodes. Hence, any node in the neighborhood of the source node will be awakened. Unlike STEM, in PTW the burden for tone detection is shifted from the receiver to the sender. This means that the duration of the wakeup tone is long enough to be detected by the receiver that turns on its wakeup radio periodically. The rationale behind this solution is that the sender only sends a wakeup tone when an event is detected, while receivers wakeup periodically. In addition, the wakeup procedure is pipelined with the packet transmission so as to reduce the wakeup latency and, hence, the overall message latency. The idea is illustrated in FIG. 10 with reference to the string topology network depicted in FIG. 9

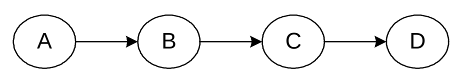


FIG. 9: String topology network.

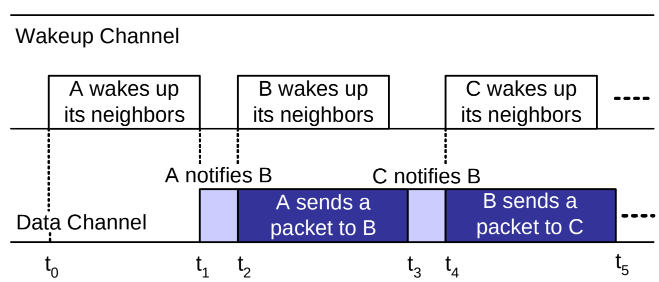


FIG. 10: Pipelined wakeup procedure in PTW.

Let's suppose that node A has to transmit a message to node D through nodes B and C. At time $t_0$ A starts the procedure by sending a tone on the wakeup channel. This tone awakens all A's neighbors. At time $t_1$ A sends a notification packet to B on the data channel to inform that the next data packet will be destined to B. Upon receiving the notification messages all A's neighbors but B learn that the following message is not intended for them. Therefore, they turn off their data radio. Instead, B realizes to be the destination of next data message, and replies with a wakeup acknowledgment on the data channel. Then, A starts transmitting the data packet on the data channel. At the same time, B starts sending a tone on the wakeup channel to awake all its neighbors. As shown in FIG. 10, the packet transmission from A to B on the data channel, and the B's tone transmission on the wakeup channel are done in parallel. As in STEM, the data transmission is regulated by the underlying MAC protocol. In [20] it is shown by simulation that, if the time spent by a sensor network in the monitoring state is greater than several minutes, PTW outperforms STEM significantly, both in terms of energy saving and message latency, especially when the bit rate of sensor nodes is low.

Both STEM and PTW assume that the power consumption of the wakeup radio is not very different from that of the data radio. Therefore, they use an asynchronous sleep/wakeup scheme for enabling a duty cycle on the wakeup radio as well. A different approach is using a low-power radio for the wakeup channel. The low-power radio is continuously in stand-by, and whenever receives a signal it wakes up the data radio [23], [24], [25], [26]. The wakeup latency is thus minimized. The main drawback of this approach is that the transmission range of the wakeup radio is significantly smaller than that of the data radio. This may limit the applicability of such a technique as a node may not be able to wakeup a neighboring node even if it is within its data transmission range. For example, in [26] the low power radio operates at 915 MHz (ISM band) and has a transmission range of approximately 332 ft in free space, while the IEEE 802.11 card operate at 2.4 GHz with a transmission range up to 1750 ft. However, the consistency between the two channels may be ensured by using static or dynamic power control.

A side effect of using a second radio for the wakeup channel is the additional power consumption which may not be negligible even when using a low-power radio. To overcome problems associated with the extra-energy consumed by the wakeup radio [21] proposes a *Radio-Triggered Power Management* scheme. The basic idea is to use the energy contained in wakeup messages (e.g., STEM-B beacon) or signals (e.g., STEM-T and PTW tones) to trigger system transitions inside the sensor node. The radio-triggered scheme, in its simplest form, is illustrated in FIG. 11. A special hardware component, a radio-triggered circuit, is used to capture the energy contained in the wakeup message (or signal), and use such energy to trigger an interrupt for waking up the node. The radio-triggered approach is significantly different than using a stand-by radio to listen to possible wakeup messages from neighboring nodes. The stand-by radio consumes energy from the node while listening, while the radio-triggered circuit is powered by the wakeup message.

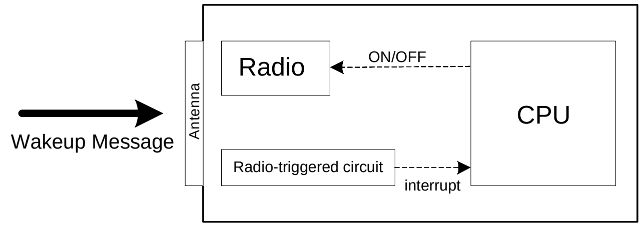


FIG. 11: Radio triggered power management.

The main drawback of the radio-triggered approach is the limitation on the maximum distance from which the wakeup message can be sent. When using the basic radio-triggered circuit illustrated above the maximum distance is 3 m. This distance may be increased up to 30 m at the cost of a more complex (and expensive) radio-triggered circuit and increased wakeup latency .

### B. Scheduled Rendezvous Schemes

Scheduled rendezvous schemes require that all neighboring nodes wake up at the same time. Typically, nodes wake up periodically to check for potential communication. Then, they return to sleep until the next rendezvous time. The major advantage of such schemes is that when a node is awake it is guaranteed that all its neighbors are awake as well. This allows sending broadcast messages to all neighbors [55]. On the flip side scheduled rendezvous schemes require nodes be synchronized in order to wakeup at the same time. Clock synchronization in wireless sensor networks is a relevant research topic. However, the discussion on clock synchronization is beyond the scope of the present chapter. Therefore, in the following we will assume that nodes are synchronized by means of some unspecified synchronization protocol.

Different scheduled rendezvous protocols differ in the way network nodes sleep and wakeup during their lifetime. The simplest way is using a *Fully Synchronized Pattern* [31]. In this case all nodes in the network wakeup at the same time according to a periodic pattern. More precisely, all nodes wakeup periodically every $T$ duration, and remain active for a fixed time $T_{active.}$ Then, they return to sleep until the next wakeup point. Due to its simplicity this sleep/wakeup scheme is used in several practical implementations including TinyDB [34] and TASK [28]. A fully synchronized wakeup pattern is also used in MAC protocols such as S-MAC [52] and T-MAC [50] (see Section VI). Even if simple, this scheme allows a low duty cycle provided that the active time ($T_{active.}$) is significantly smaller than the wakeup period $T$. A further improvement can be achieved by allowing nodes to switch off their radio when no activity is detected for at least a timeout value [50]. In addition, due to the large size of the active and sleeping part, it does not require very precise time synchronization [56]. The main drawback is that all nodes become active at the same time after a long sleep period. Therefore, nodes try to transmit simultaneously, thus causing a large number of collisions. In addition, the scheme is not very flexible since the size of wakeup and active periods is fixed and does not adapt to variations in the traffic pattern and/or network topology.

The fully synchronized scheme applies equally well to both flat and structured sensor networks. To this end it may be worthwhile recalling that many routing protocols superimpose a tree or cluster-tree organization to the network by building a data gathering tree (or routing tree) typically rooted at the sink node. Some sleep/wakeup schemes take advantage of the internal network organization by sizing active times of different nodes according to their position in the data gathering tree. The latter could change over time due to node failures, topology changes (node that joins or leaves), etc. In addition, it could be recomputed periodically by the routing protocol to achieve load balancing among nodes. However, under the assumption that nodes are static, it can be assumed that the data gathering tree remains stable for a reasonable amount of time [33].

In the *Staggered Wakeup Pattern*, shown in FIG. 12, nodes located at different levels of the data gathering tree wakeup at different times. Obviously, the active parts of nodes belonging to adjacent levels must be partially overlapping to allow nodes to communicate with their children. Finally, the active parts of different levels are arranged in such way that the portion of active period a node uses to receive packets from its children is adjacent to the portion it uses to send packet to its parent (FIG. 12). This minimizes the energy dissipation to transitioning from sleep to active mode.

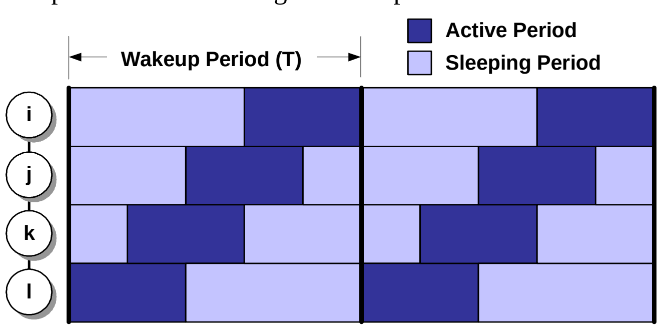


FIG. 12: Staggered sleep/wakeup pattern.

The staggered wakeup pattern shown in FIG. 12 is also called backward staggered pattern [31] as it optimizes packet latency in the backward direction i.e., from leaf nodes to the root (which is typically the sink node). It is also possible to arrange nodes' active periods in such way to optimize the forward packet latency (i.e., from the root to leaves). The resulting scheme, called forward staggered pattern [31] is however not very used in practice, because in real networks most of data flows from sensor nodes to the sink. A combination of the backward and forward staggered pattern is also possible (see below).

The (backward) staggered scheme was first proposed in the framework of TinyDB [34] and TAG [35]. Due to its nice properties this scheme has been then considered and analyzed in several other papers ([29], [33], [32], [36] among others) even if with different names. A staggered wakeup pattern is also used in D-MAC [33] (see Section VI).

With respect to the fully synchronized scheme the staggered scheme has several advantages. First, since nodes at different levels of the data gathering tree wakeup at different times, at a given time only a (small) subset of nodes in the network will be active. Thus, the number of collisions is potentially lower as there are less nodes that contend for channel access (assuming that a contention-based MAC protocol is used). For the same reason the active period of each node can be significantly shortened with respect to the fully synchronized scheme, thus resulting in energy saving. This scheme is also suitable to data aggregation. Parent nodes receive data from all their children before they forward such data to their own parent at the higher level. This allows parent nodes to filter data received from children, or to aggregate them with their own data.

The staggered scheme has some drawbacks in common with the fully synchronized scheme. First, since nodes located at the same level in the data gathering tree wakeup at the same time, collisions can potentially still occur. In addition, this scheme has limited flexibility due to the fixed duration of the active ($T_{active}$) and wakeup ($T$) periods. The active period is often the same for all nodes in the network. For example, in [35] $T_{active.}$ is set to the duration of the wakeup period $T$ divided by the maximum number of hops in the data gathering tree, while in [39] it is based on the delay to traverse a single hop.

Ideally, the active period should be as low as possible, not only for energy saving but also for minimizing the latency experienced by packets to reach the root node (see FIG. 12). In addition, since nodes located at different levels of the data gathering tree manage different amounts of data, active periods should be sized based on individual basis. Finally, even assuming static nodes, topology changes and variations in the traffic patterns are still possible. The active period of nodes should thus adapt dynamically to such variations.

An adaptive and low latency staggered scheme is proposed in [27] (a somewhat similar approach is also taken in [33]). By setting the length of the active period to the minimum value consistently with the current network activity, this adaptive scheme not only minimizes the energy consumption but also provides a lower average packet latency with respect to a fixed staggered scheme. In addition, by allowing different length of the active period for nodes belonging to the same level but associated with different parents, it also reduces the number of collisions [27].

Another adaptive scheme is the *Flexible Power Scheduling* (FPS) proposed in [30]. FPS takes a slotted approach, i.e. time is assumed to be divided in slots of duration $T_s$. Slots are arranged to form periodic cycles, where each cycle is made up of *m* slots and has a duration of $T_c = m\, T_s$. Each node maintains a power schedule of what operations it performs during a cycle. Obviously, a node must keep its own radio on only when it is has to receive/transmit from/to other nodes. Slotted schemes typically suffer from two common problems: they are not flexible and require a strict synchronization among nodes. To overcome the lack of flexibility FPS includes a on-demand reservation mechanism that allows nodes to reserve slots in advance. As far as synchronization, slots are relatively large so that only coarse-grain synchronization is required.

Several other sleep/wakeup scheme that still leverage the tree network organization have been considered and analyzed [32], [57]. The *Shifted Even and Odd Pattern* is derived from the Fully Synchronized Pattern by shifting the wakeup times of nodes in even levels by *T/2* (*T* being the wakeup period). This minimizes the *overall* average packet latency i.e., the average latency considering both the forward and backward directions, and also increases the network lifetime. Finally, the *Two-Staggered Pattern* and *Crossed Staggered Pattern* [31] are obtained as combinations of the of the Backward Wakeup Pattern and Forward Wakeup Pattern.

In [31] the authors also propose a multi-parent scheme which can be combined with any of the above sleep/wakeup patterns. The multi-parent scheme assigns multiple parents (with potentially different wakeup pattern) to each node in the network. This results in significant performance improvements in comparison with single-parent schemes.

### C. Asynchronous Schemes

Asynchronous schemes avoid the tight synchronization among network nodes required by scheduled rendezvous schemes. They allow each node to wakeup independently of the others by guaranteeing that neighbors always have overlapped active periods within a specified number of cycles.

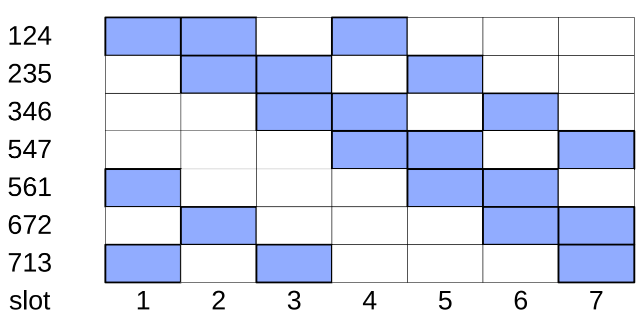


FIG. 13: An example of asynchronous schedule based on a symmetric (7,3,1)-design of the wakeup schedule function.

Asynchronous wakeup was first introduced in [58] with reference to IEEE 802.11 ad hoc networks. The basic IEEE 802.11 Power Saving Mode (PSM) [59] has been conceived for single-hop ad hoc network and thus it is not suitable to multi-hop ad hoc networks where nodes may also be mobile. In [58] the authors propose three different asynchronous sleep/wakeup schemes that require some modifications to the basic PSM.

More recently, Zheng et al. [44] took a systematic approach to design asynchronous wakeup mechanisms for ad hoc networks. Their scheme applies to wireless sensor networks as well. They formulate the problem of generating wakeup schedules that rely upon asynchronous wakeup mechanisms as a block design problem and derive theoretical bounds under different communication models. Based on the optimal results obtained from the block design problem, they design an *Asynchronous Wakeup Protocol* (AWP) that can detect neighboring nodes in a finite time without requiring slot alignment. The proposed asynchronous protocol is also resilient to packet collisions and variations in the network topology. The basic idea is that each node is associated with a Wakeup Schedule Function (WSP) that is used to generate a wakeup schedule. For two neighboring nodes to communicate their wakeup schedules have to overlap, regardless of the difference in their clocks. The idea is illustrated in FIG. 13 by means of an example of asynchronous wakeup schedule for a set of 7 neighboring nodes. This example is based on a symmetric (7,3,1)-design of the wakeup schedule function. Symmetric means that all nodes have the same duty cycle, while (7,3,1)-design indicates that: (**i**) each schedule repeats every 7 slots; (**ii**) each schedule has 3 active slots out of 7 (blue slots); and (**iii**) any two schedules overlap for at most 1 slot. As shown in FIG. 13, by following its own schedule (i.e., by turning on the radio only during its active slots) each node is guaranteed to communicate with any other neighboring node.

The above scheme ensures that each node will be able to contact any of its neighbors in a finite amount of time. However the packet latency introduced may be heavy especially in multi-hop networks. In addition, it never happens that all neighbors are simultaneously active. Therefore, it is not possible to broadcast a message to all neighbors [55].

*Random Asynchronous Wakeup* (RAW) [42] takes a different approach as it leverages the fact that sensor networks are typically characterized by a high node density. This allows the existence of several paths between a source and a destination and, thus, a packet can be forwarded to any of such available paths. Actually, the RAW protocol consists of a routing protocol combined with a random wakeup scheme. The routing protocol is a variant of geographic routing. While in geographic routing a packet is sent to a neighbor that is closest to the destination, in RAW the packet is sent to any of the active neighbors in the *Forwarding Candidate Set*, i.e., the set of active neighbors that meet a pre-specified criterion. The basic idea of the random wakeup scheme is that each node wakes up randomly once in every time interval of fixed duration *T*, remains active for a predefined time $T_a$ ($T_a \leq T$), and then sleeps again. Once awake, a node looks for active neighbors by running a neighbor discovery procedure. If there are *m* neighbors in the forwarding set of node *S* to which a packet destined to node *D* can be transmitted, then the probability that at least one of such nodes is awake, when S is awake, is given by

$$P = 1 - \left(1 - \frac{2 \cdot T_a}{T}\right)^m \quad (2)$$

If the sensor network is dense, the number (*m*) of neighbors in the *Forwarding Candidate Set* is large and, by (2), the probability *P* to find an active neighbor to which forward the packet is large as well.

The random wakeup scheme is extremely simple and relies only on local decisions. This makes it well-suited for networks with frequent topology changes. On the other side, it is not suitable for sparse networks. When a node wakes up in RAW it is not sure to find another active neighbor, even if it is very likely thanks to the network density. Therefore, RAW does not guarantee the packet forwarding within one time frame (*T*), while AWP does.

An alternative approach to ensure that an asynchronous node – typically a sender – finds its communication counterpart (i.e., the receiver) active when it wakes up, is forcing the receiver to listen periodically. The receiver wakes up periodically and listens for a short time to discover any potential asynchronous sender. If it does not detect any activity on the channel it returns to sleep, otherwise remains active to send/receive packets. Even if the receiver need to periodically wakeup this scheme falls in the category of asynchronous schemes because nodes do not need to be synchronized.

Two different variants are possible to discover asynchronous senders by periodic listening. We have already introduced these two variants with reference to STEM-B and PTW, respectively. However, their usage is more general. This is why we re-discuss them in this context.

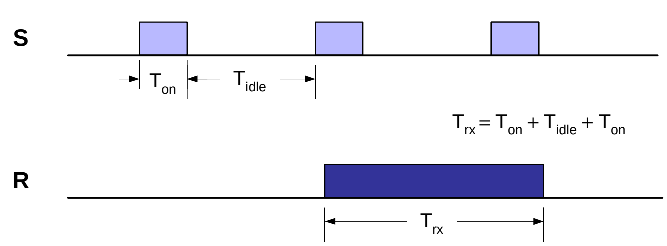


FIG. 14: Discovery of an asynchronous sender through periodic listening. The sender transmits a stream of periodic discovery messages.

In the first variant, depicted in FIG. 14 the asynchronous sender transmits a stream of periodic discovery messages (e.g., STEM-B beacons [22]). As anticipated in Section IV-A, to ensure the correct discovery of the sender, the receiver's listening time ($T_{rx}$) must be at least equal to $T_{on}$+ $T_{idle}$+ $T_{on}$, where, $T_{on}$ is the time for transmitting a discovery message and $T_{idle}$ is the time between the end of a discovery message and the start of the next one.

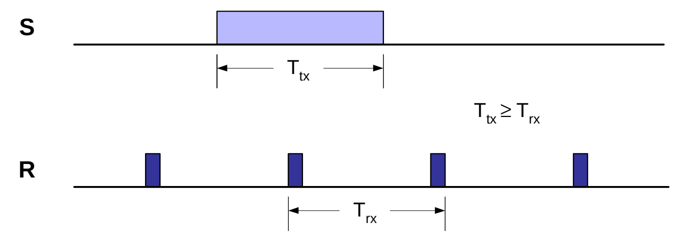


FIG. 15: Discovery of an asynchronous sender through periodic listening. The sender transmits a single long discovery message.

The second variant is illustrated in FIG. 15 and differs from the previous one in that the sender transmits a single long discovery message instead of a stream of periodic discovery messages. In this case the receiver listening time can be very short provided that the duration of the discovery message ($T_{tx}$) is, at least, equal to the listening period $T_{rx}$. This variant is used for enabling duty cycling on the wakeup channel in PTW. A similar scheme is also used in B-MAC [43] (see Section VI-B). In addition, both variants are very suitable for sensor networks where mobile nodes (data mules) are used to collect data [60], [61]. Since the mule arrival time is usually unpredictable, static nodes typically use an asynchronous scheme, like the ones shown in FIG. 14 and FIG. 15, for mule discovery. This allows the timely discovery of the nearby mule without keeping the radio continuously on [60].

## VI. MAC PROTOCOLS WITH LOW DUTY CYCLE

Several MAC protocols for wireless sensor networks have been proposed in the literature. Most of them implement a low duty-cycle scheme for power management. We will survey below the most common MAC protocols by classifying them according to the taxonomy introduced in Section III-B. Other previous surveys and introductory papers on MAC protocols for wireless sensor networks are also available in the literature (see, for example, [62], [63] and [64]). In the following discussion we will focus mainly on power management issues rather than on channel access methods.

### A. TDMA-based MAC Protocols

In TDMA-based MAC protocols [45], [65], [66], [46], [47] time is divided in (periodic) frames and each frame consists of a certain number of time slots. Every node gets assigned to one or more slots per frame, according to a certain scheduling algorithm, and uses such slots for transmitting/receiving packets to/from other nodes. In many cases nodes are grouped to form clusters with a cluster-head which is in charge to assign slots to nodes in the cluster (as in Bluetooth [65], LEACH [66], and Energy-aware TDMA-based MAC [45]).

TRAMA [47] is a TDMA-based and energy-efficient channel access scheme for sensor networks. TRAMA divides time in two portions, a random-access period and a scheduled access period. The random access period is devoted to slot reservation and is accessed with a contention-based protocol. On the contrary, the scheduled access period is formed by a number of slots assigned to an individual node. The slot reservation algorithm is the following. First, nodes obtain two-hop neighborhood information, which are required to establish collision free schedules. Then, nodes start an election procedure to associate each slot with a single node. Every node gets a priority of being the owner of a specific slot. This priority is calculated as a hash function of the node identifier and the slot number. The node with the highest priority becomes the owner of a given slot. Finally, nodes send out a synch packet containing a list of intended neighbor destinations for subsequent transmissions. Thanks to this information, nodes can agree on the slots which they must be awake in. Unused slots can be advertised by their owners for being re-used by other nodes.

TDMA-based protocols naturally enable duty cycling as nodes turn on their radio only during their own slots and sleep for the rest of the time. By an appropriate design of the slot assignment algorithm, and a correct sizing of the protocol parameters, it is thus possible to minimize energy consumption. In addition, TDMA-based MAC protocols can easily solve (i.e., without extra message overhead) problems associated with interference among nodes (e.g., the hidden node problem) as it is possible to schedule transmissions of neighboring nodes to occur at different times.

On the other side, TDMA MAC protocols have several drawbacks that limit their usage in real sensor networks [48]. First, they lack *flexibility*. In a real sensor network there may be frequent topology changes caused by time-varying channel conditions, physical environmental changes, nodes that run out of energy, and so on. Handling topology changes in an efficient way is hard and may require a global change in the slot allocation pattern. Second, TDMA schemes have limited *scalability*. Finding an efficient time schedule in a scalable fashion is not trivial. In many cases (e.g., in Bluetooth [65] or LEACH [66]) a central node is required to schedule channel access in a collision-free manner. Third, TDMA MAC protocols require tight *synchronization* among network nodes which introduces overhead in terms of control message exchange and, thus, additional energy consumption. Fourth, finding an *interference-free* schedule is a very hard task since interference ranges are typically larger than transmission ranges, i.e., many network nodes may interfere even if they are not in the transmission range of each other [49]. Therefore, a slot assignment based on transmission ranges is not, very likely, an interference-free schedule. In addition, interference ranges are time-varying which makes static slot assignment unsuitable for real environments. On the other hand, adapting the schedule to varying external conditions is not trivial. Fifth, under low traffic conditions, TDMA MAC protocols perform worse than CSMA MAC protocols both in terms of *channel utilization* and *average packet delay*. This is because in TDMA schemes nodes have to wait for their own slots to transmit while in CSMA schemes node can try channel access at any time and access is almost immediate as there is low contention.

For all the above reasons, TDMA MAC protocols are not very frequently used in practical wireless sensor networks.

### B. Contention-based MAC Protocols

Most of MAC protocols proposed for wireless sensor networks are contention-based protocols.

B-MAC (Berkeley MAC) [43] is a low complexity and low power MAC protocol developed at UCB, and shipped with the TinyOS operating system [67]. The goal of B-MAC is to provide a few core functionalities and an energy efficient mechanism to access the channel. First, B-MAC implements a few basic channel access control features: a backoff scheme, an accurate channel estimation facility and optional acknowledgements. Second, to achieve a low duty cycle B-MAC uses an asynchronous sleep/wake scheme based on periodic listening (see Section V-C) called Low Power Listening (LPL). Nodes wake up periodically to check the channel for activity. The wakeup time is fixed while the check interval can be specified by the application. The ratio between the wake interval and the check interval defines the node duty cycle. B-MAC packets consist of a long preamble and a payload. The preamble duration is at least equal to the check interval so that each node can always detect an ongoing transmission during its check interval. This approach does not require nodes to be synchronized. In fact, when a node detects channel activity, it just receives the preamble and then the payload.

S-MAC (Sensor-MAC) [52] is a duty-cycle based MAC protocol for multi-hop sensor networks proposed by researchers at UCLA. Nodes exchange *sync* packets to coordinate their sleep-wakeup periods. Every node can follow its own schedule or follow the schedule of a neighbor. A node can eventually follow both schedules if they do not overlap. Nodes using the same schedule form a virtual cluster. The channel access time is split in two parts. In the listen period nodes exchange sync packets and special control packets for collision avoidance (in a similar way to the IEEE 802.11 standard [59]). In the remainder period the actual data transfer takes place. The sender and the destination node are awake and talk each other. Nodes not concerned with the communication process can sleep until the next listen period. To avoid high latencies in multi-hop environments S-MAC uses an adaptive listening scheme. A node overhearing its neighbor's transmissions wakes up at the end of the transmission for a short period of time. If the node is the next hop of the transmitter, the neighbor can send the packet to it without waiting for the next schedule. The parameters of the protocol, i.e. the listen and the sleep period, are constants and cannot be varied after the deployment.

T-MAC (Timeout MAC) [50] is an enhancement of S-MAC designed for variable traffic load. In detail, T-MAC employs a synchronization scheme based on virtual clusters similar to S-MAC's. Schedules between nodes define frames within communication takes place. Queued packets are transmitted at the beginning of the frame in a burst. Between bursts nodes can go to sleep to save energy. The active time is defined on the basis of an activation period, in order to reduce the amount of idle listening and adapt to traffic as well. A node can go to sleep if no significant event (e.g. the reception of a packet, overhearing of RTS/CTS etc.) has occurred for the duration of the activation period. The length of the activation period must be chosen carefully to avoid the early-sleeping problem. In fact a node can go to sleep when a neighbor has still messages for it. This happens, for example, when the communication pattern is asymmetric. T-MAC provides some mechanisms to reduce the early sleeping problem. They also help in the sensor networks multi-hop communication pattern, where the nodes close to the sink have to handle more traffic. Besides, T-MAC uses explicit signaling to reduce the sleep latency. By using special control packets, nodes can hear the intention of another node to send a packet, so that they can awake to receive it. T-MAC has better values of energy efficiency and latency than S-MAC.

D-MAC [33] is an adaptive duty cycle protocol optimized for data gathering in sensor networks where a tree organization has been established at the network layer. Although duty-cycle based MAC protocols are energy efficient, they suffer sleep latency, i.e. a node must wait until the receiver wakes up before it can forward a packet. This latency increases with the number of hops. In addition, the data forwarding process from the nodes to the sink can experience an interruption problem. In fact, the radio sensitivity limits the overhearing range, thus nodes outside the range of the sender and the receiver can't hear the ongoing transmission and go to sleep. That's why in S-MAC and T-MAC the data forwarding process is limited to a few hops. In DMAC, instead, the nodes' schedules are staggered according to their position in the data gathering tree, i.e., nodes' active periods along the multi-hop path are adjacent in order to minimize the latency. Each node has a slot which is long enough to transmit a packet. A node having more than one packet to transmit explicitly requests additional slots to their parent. In this way the length of the active periods can be dynamically adapted to the network traffic. Finally, D-MAC uses a data prediction scheme to give all children the chance to transmit their packets.

IEEE 802.15.4 [51] is a standard for low-rate, low-power *Personal Area Networks* (PANs). A PAN is formed by one PAN coordinator and, optionally, by one or more coordinators. The other nodes must associate with a (PAN) coordinator, who manages the communication within the network. Supported network topologies are star (single-hop), cluster-tree and mesh (multi-hop). The IEEE 802.15.4 standard supports two different channel access methods: a *beacon enabled* mode and a *non-beacon enabled* mode. The *beacon enabled mode* provides an energy management mechanism based on a duty cycle. Specifically, it uses a superframe structure which is bounded by *beacons* – special synchronization frames generated periodically by coordinator nodes. Each superframe consists of an *active period* and an *inactive period*. In the active period devices communicate with the coordinator they associated with. The active period can be further divided in a contention access period (CAP) and a collision free period (CFP). During the CAP a slotted CSMA/CA algorithm is used for channel access, while in the CFP a number of guaranteed time slots (GTSs) can be assigned to individual nodes. During the inactive period devices enter a low power state to save energy. In the *non-beacon enabled* mode there is no superframe structure, i.e., nodes are always in the active state and use an unslotted CSMA/CA algorithm for channel access and data transmission.

IEEE 802.15.4 beacon-enabled mode is suitable for single-hop scenarios. However, the beacon-based duty-cycle scheme have to be extended for multi-hop networks. In [36] the authors propose a maximum delay bound wakeup scheduling specifically tailored to IEEE 802.15.4 networks. The sensor network is assumed to be organized as a cluster tree. An optimization problem is formulated in order to maximize network lifetime while satisfying latency constraints. The optimal operating parameters for single coordinators are then obtained. Therefore, an additional extended synchronization scheme is used for inter-cluster communication.

Contention-based MAC protocols are robust and scalable. In addition, they generally introduce a lower delay than TDMA-based MAC protocols and can easily adapt to traffic conditions. Unfortunately, their energy expenditure is higher than TDMA MACs because of collisions and multiple access schemes. Duty-cycle mechanisms can help reducing the energy wastage, but they need to be designed carefully to be adaptive and low latency.

### C. Hybrid MAC Protocols

Hybrid MAC protocols [53], [48] try to combine the strength of TDMA-based and CSMA-based MAC protocols, while offsetting their weaknesses. The idea of switching the protocol behavior between TDMA and CSMA, depending on the level of contention, is not new. In [53] the authors propose an access scheme for a WLAN environment that relies upon a *Probabilistic TDMA* (PTDMA) approach. In PTDMA time is slotted, and nodes are distinguished in *owners* and *non-owners*. The protocol adjusts the access probability of *owners* and *non-owners* depending on the number of senders. By doing so it adapts the MAC protocol to work as a TDMA or CSMA scheme depending on the level of contention in the network.

However, PTDMA was conceived for a one-hop wireless scenario. Therefore, it does not take into account issues such as topology changes, synchronization errors, interference irregularities which are quite common in wireless sensor networks.

Z-MAC [48] is a hybrid protocol specifically designed for sensor networks. The protocol includes a preliminary setup phase during which the following operations are carried out: *neighbor discovery*, *slot assignment*, *local frame exchange*, and *global time synchronization*. By means of the neighbor discovery process each node builds a list of two-hop neighbors. This list is then used by a distributed slot assignment algorithm to assign slots to every node in the network. This algorithm guarantees that no two nodes in the two-hop neighborhood are assigned to the same slot. In other words it guarantees that no transmission from a node to any of its one-hop neighbor interferes with any transmission from its two-hop neighbors. The local frame exchange is aimed at deciding the *time frame*. Z-MAC does not use a global frame equal for all nodes in the network. It would be very difficult and expensive to adapt when a topology change occurs. Instead, Z-MAC allows each node to maintain its own local time frame that depends on the number of neighbors and avoids any conflict with its contending neighbors. Finally, the global time synchronization process is aimed at synchronizing all nodes to a common clock. The local slot assignment and time frame of each node are then forwarded to its two-hop neighbors. Thus any node has slot and frame information about any two-hop neighbors and all synchronize to slot 0. At this point the setup phase is over and nodes are ready for channel access, regulated by the transmission control procedure. Nodes can be in one of the following modes: *Low Contention Level* (LCL) and *High Contention Level* (HCL). A node is in the LCL unless it has received an *Explicit Contention Notification* (ECN) within the last $T_{ECN}$ period. ECNs are sent by nodes when they experience high contention. In HCL only the owners of the current slot and their one-hop neighbors are allowed to compete for accessing the channel. In LCL any node (both owners and non-owners) can compete to transmit in any slot. However, the owners have priority over non-owners. This way Z-MAC can achieve high channel utilization even under low contention because a node can transmit as soon as the channel is available.

# VII. CROSS-LAYER DESIGN

Even though energy conservation is a general concern for all mobile computing fields, it is probably the driving force in wireless sensor networks. Researchers in this field tend to look at low energy consumption as the main target, and trade off any other performance figure (e.g., throughput, delivery ratio, reliability) for longer lifetime. This approach naturally leads to optimize the network protocols design as much as possible from an energetic standpoint. The clean separation (and interfaces) between layers of traditional protocol stacks is often abandoned, because protocol designers need to gather information from any layer, provided it is useful to make the protocol more energy-efficient. Cross layering in wireless sensor networks is so common, that sometimes papers' authors neglect to mention that their protocol exploits cross-layer interactions.

Broadly speaking, we can categorize papers adopting cross layering for energy conservation in sensor networks in three classes: *algorithmic* approaches, *side-effect* approaches, and *pure cross-layer energy-conservation schemes*. In the following of this section we will separately survey each class. Finally, we will highlight some architectural issues related to cross-layering in Section D.

## A. Algorithmic Approaches

Papers falling in this class abstract the problem of increasing the sensor network lifetime through optimization programming techniques. The typical framework consists in defining an (possibly linear) optimization problem, in which some function of the network energy consumption has to be minimized (or, equivalently, the network lifetime has to be maximized). The constraints of the problem allow to model real constraints of the network. From a networking perspective, these formulations are cross-layer in nature, since the parameters of the objective function and the constraints usually depend on data that resides at different layers of the stack. For example, in [68] the authors focus on sensor networks supporting in-network aggregation for distributed queries. Specifically, the network has to deliver data to the sink in order to answer queries in which aggregate operators can be specified. Aggregation is not performed at the sink on the raw data sensed from the environment, but is computed in the network in a distributed and incremental fashion, so as to reduce the traffic (and the energy consumption). Authors define optimization problems to find the optimal routing policy in terms of energy consumption. In other words, the solution of the problem is the routing policy that achieves the minimum energy consumption for the given network. A cross-layer feature of this particular example is the fact that different problem formulations are used depending on the type of aggregate queries taken into consideration. So, the routing policy is actually computed based on application-level information, i.e., the kind of query submitted to the network. The work presented in [69] also falls in this category. In this case, authors jointly optimize (in terms of energy consumption) the topology control, the routing, and the sleep/wakeup schedule of the nodes based on the physical data rate the network is operating in. A further example of this approach is presented in [70]. Specifically, the authors define optimization problems that provide the optimal parameters in terms of energy consumption for the transmit power levels, the routing flow, and the links' scheduling. The same approach is also taken in

[71], even though the focus there is specifically on UWB sensor networks. Other examples can be found in the Related Work section of [70].

Usually, the optimization problems defined in this way turn out to be NP-complete. After proving this, authors define heuristics that are able to approximate the optimal solution with a certain (hopefully small) bound.

Even though such approaches are interesting from an intellectual standpoint, and also provide solid analytical frameworks, they tend to be very abstracted from the real world. Drastic approximations are usually necessary to make the problem analytically tractable. But the serious drawback is the fact that it is typically very difficult to guess how much these approximations will impact on the performance of a real system.

### B. Side-effect Approaches

Papers in this class usually do not share the same drastic approximations used by algorithmic approaches, and do not deal with the energy management problem via optimization problems. Instead, they propose energy-aware networking protocols. We name this class as Side-Effect Approaches, because the main focus of such papers is not on designing cross-layer energy management schemes. Rather, they design cross-layer networking protocols that, as a side effect, also turn out to reduce the energy consumption with respect to other reference cases.

There are plenty of papers following this approach in the literature. Just to give some examples, we focus on [72], [73], [74], and [75]. The authors of [72] define an energy-aware routing protocol that selects routes based on (**i**) the link error rates, and (**ii**) the end-to-end reliability requirement of the data to be routed. The claim of [72] is that, in order for routing policies to be energy efficient, it is not sufficient to take into account just single-link qualities, because data have to be forwarded over multi-hop paths. Thus, it is better to estimate the routes cost based on the expected total time required to reach the destination. This quantity is clearly dependent on the reliability scheme used by the application (e.g., end-to-end or hop-by-hop).

In [73] authors propose an energy-efficient protocol to disseminate data from sensor nodes to multiple sinks. The novelty of this paper is that the dissemination tree is built based on the nodes' locations and on the packet traffic rates among nodes.

As [72], both [74], [75], and [76] define energy-aware routing protocols. But, with respect to [72], they take a quite novel approach. Specifically, they assume that the sink could be mobile, and jointly identify the best sink mobility pattern and routing policy for sensor nodes to reach the sink that minimizes the energy consumption of the network (or, equivalently, that maximizes the network lifetime).

### C. Pure Cross-layer Power-Management Schemes

With respect to papers that achieve energy conservation as a side effect, papers in this category directly aim to design energy management schemes, by exploiting information residing at different layers of the network stack. To make the difference clearer, an energy conservation scheme has to care about the energy consumed by the sensor nodes (or, better yet, by the sensor network) in all possible operating conditions. For example, an energy-aware routing protocol can optimize the forwarding procedure, but cannot manage the energy spent by sensor nodes when they are not forwarding anything. Of course such approaches are not mutually exclusive in principle.

The work in [77] proposes a power management scheme that turns off the wireless transceiver of sensor nodes when they are not required by the running applications. More specifically, it assumes a TDMA MAC protocol, and defines the TDMA schedule based on the application demands. Under the assumption of applications periodically reporting to sinks, MAC-level frames are aligned with the beginning of reporting periods. Abstracting a bit from [77], we can envision cross-layer energy managers that switch on and off the networking subsystem of sensor nodes based on the demand of all networking layers.

The definition of sleep/wakeup patterns is the goal of [31], as well. Differently from standard approaches, in which a node is bound to follow a well-defined schedule, in this paper nodes can dynamically decide to join different available schedules based on the expected delay towards the destination. Essentially, when a node has to send (or forward) a packet, it chooses the schedule of the next hop corresponding to the path achieving the fastest delivery. In this case, the energy manager exploits topological information in order to decide when to turn the wireless interface on and off.

The final example we consider for this class is [78]. Also in this case authors focus on a sensor network in which sensors have to periodically report to a sink. The main idea of this paper is exploiting the temporal correlation of physical quantities (e.g., temperature) to reduce the amount of time the nodes has to turn their wireless interface on. At the same time, this energy manager takes also into consideration the maximum inaccuracy that the application is willing to tolerate on reports, and the maximum delay that the application can admit in starting reporting. Based on the samples collected from the environment, each sensor node computes a model of future readings. This model is sent to some node responsible for storing models. This node is then responsible for generating reports (and sending them to the sink) on behalf of the sensor node. While the model is accurate enough, the sensor nodes can keep its wireless interface off. Readings that differ from the predicted values by some application-defined threshold triggers a violation. Only in this case the sensor node turns the wireless interface on and sends the actual reading to the sink. The sink sends new queries not directly to the interested sensor node, but to the same node responsible for storing the models, where they are temporarily buffered. Sensor nodes are required to periodically poll this node to check for possible new queries. The polling period is based on the maximum delay the application is willing to tolerate.

### D. Architectural Issues

Despite its indubitable advantages, cross layering is a tool to be handled with some care. A recent paper by Malesci and Madden brought this out very clearly [56]. Authors highlight through experimental measures that the performance of a protocol in a given layer depends on "hidden" cross-layer interactions with protocols in other layers. For example, they show drastic performance differences when the same MAC protocol is used with different routing protocols. Such results are not very surprising per se, but are very significant in the context of wireless sensor networks. Actually, a flurry of protocols for any layer have been proposed for sensor networks, and no one is nowadays a clear winner. Thus, performance evaluations should carefully state the limits of their validity, since changes in any layer of the stack might significantly impact on the performance of any other layer. Another caveat from [56] is the fact that cross-layering might result in monolithic network stack in which layers are coupled so tightly that any maintenance or partial replacement becomes practically unfeasible. Authors note that this trend has produced vertically integrated network stacks that cannot be integrated in any way, nor can be mixed. Previously, Kawadia and Kumar raised similar concerns with respect to ad hoc networks in general [79]. To avoid such "spaghetti-like" network stacks, authors of [56] advocate the definition of standard APIs to implement cross-layer interactions.

An example of such solution is the Sensor network Protocol (SP) proposed in [80] and [81]. SP is an intermediate layer between the MAC and the network layer. SP aims to join the advantages of cross-layer optimizations and the portability of legacy-Internet solutions. It takes the footsteps of the IP protocol, in the sense that it abstracts all details of the underlying MAC protocol, while providing a standard, well-defined interface to the network layer. However, while the IP protocol is completely opaque, as it does not expose any lower-level information to above layers, SP is translucent. Specifically, it allows the network layer to gather information about the lower levels, thus enabling cross-layer optimizations. The definition of a standard interface between SP and the adjacent layers avoids spaghetti-like stacks, and improves management and portability.

Independently from the work described in [80] and [81], similar conclusions have been drawn within the MobileMAN Project [82]. In this project the focus was on mobile ad hoc networks (MANETs) rather than on sensor networks. However, the main architectural framework designed within this project could be ported to sensor networks, as well. Specifically, MobileMAN researchers have defined a NeSt (Network Status) module to implement cross-layer interaction among protocols at any layer in the stack. NeSt acts as a mediator between two protocols willing to interact. Instead of interacting directly, protocols generate information that is stored by the NeSt (e.g., the link layer could ask the NeSt to store the packet-drop probability), and query the NeSt to get information generated by other protocols (e.g., the transport protocol may wish to get a notification when a link breaks). Interactions with the NeSt occur through a well-defined API, which actually shields and insulates protocols from each other. Even though the NeSt and SP definitions appear to have come out in parallel, NeSt extends the concept of translucency between protocols to any layer in the stack, instead of confining it between the MAC and the routing layers.

In conclusion, we believe that cross-layering is actually the way to go to implement energy-efficient networking schemes in sensor networks. Indeed, the advantages brought by cross-layering are really huge. However, we agree that cross-layering has to be implemented without breaking stacks maintainability and portability. Approaches like SP and NeSt look like the right direction to pursuit.

## VIII. ENERGY-EFFICIENT NETWORKING PROTOCOLS

Networking protocols for sensor networks have been extensively studied and constitute a large part of the research activity on sensor networks. The interested readers can find an excellent and comprehensive coverage of this topic in [3] and [83]. Below, we will briefly discuss issues related to energy conservation. Specifically, we will survey how energy efficiency can be achieved at different layers of the OSI reference model. In fact, energy conservation is a cross-layer issue and should be implemented at each layer of the protocol stack.

### A. Physical and Data Link Layers

For Physical and Data Link layers the power efficiency questions are similar to those addressed in wireless networks: how to transmit in a power efficient way bits and frames, respectively, to devices one-hop away. Apart from medium access control, discussed in Section VI, these problems include identifying suitable modulation schemes, efficient FEC strategies, etc. (see [3], [84]). Of course, the solutions of these problems are strongly affected by the sensor-device resources' constraints. The proposed solutions are generally independent from the applications, however, recently some authors [85] proposed to apply data-centric policies also at the MAC layer. The basic idea is to exploit spatial correlation among neighboring nodes to reduce the number of transmissions at the MAC layer.

### B. Network layer

Many solutions have been proposed in the literature for energy efficient routing in wireless sensor networks. A comprehensive presentation of this topic can be found in [86] and [87]. A taxonomy of routing protocols for wireless sensor networks is shown in FIG. 16. Almost all routing protocols for sensor networks can be classified by means of the network structure they exploit. These *network-structure-based* protocols can be further divided in three categories: location-based, hierarchical and flat. Some other protocols, however, do not fit this scheme and are generally distinguished on the basis of their operations. For example, [88] and [89] setup routes as the solution of a network flow model. Furthermore, SAR [90] and SPEED [91] use QoS metrics to trade off energy consumption and data quality. The work in [92] defines routes based on reliability achieved via a separate link level mechanism, also defined in the paper. Finally, the work in [93] focuses on the funneling effect, i.e., the fact that nodes close to sink(s) tend to exhaust their energy more quickly than nodes far away, because they have to route more traffic towards the sink(s). To fight this problem, authors define the optimal transmission range of nodes depending on their distance (in terms of hops) from the sink. The rationale is to reduce the transmission power of nodes close to the sink so as to balance the additional burden they have to carry due to routing tasks.

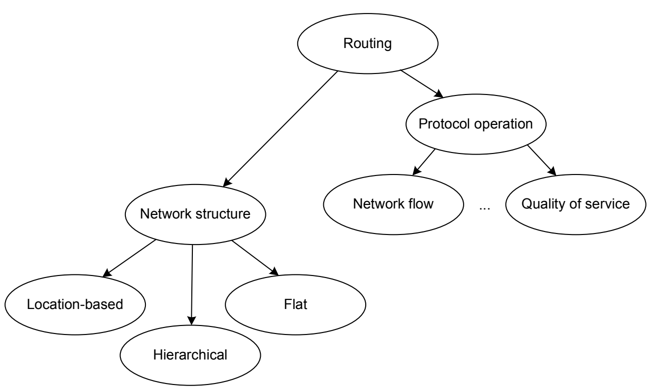


FIG. 16: Taxonomy of routing protocols.

In the following we will focus on network-structure-based protocols because they are the most representative from the energy-aware design perspective.

*Location-based routing protocols* exploit nodes' position or proximity to route data in the network. Many of these protocols – e.g. GAF [15], SPAN [17] and ASCENT [16] – also use location information to power off the nodes which are not involved in the routing process. From this point of view they can also be seen as topology control protocols, as explained in Section IV. Nevertheless, some protocols take different approaches. For example, GEAR [94] splits the forwarding process in two steps: forwarding toward the target region and forwarding within the target region. The first step uses an estimated cost based on nodes' distance and residual energy. The second step involves a combination of geographic forwarding and restricted flooding. Another protocol exploits low-power GPS receivers to obtain the so called Minimum Energy Communication Network (MECN) [95]. This protocol builds a graph which accounts for the power consumption needed to transmit or receive packets. Once this graph is available, a distributed algorithm compute the minimum energy subnetwork that can be used for communications. An extension of this protocol can find the Smallest MECN, with higher energy gains if the broadcast region is circular around the broadcast transmitter [96].

*Hierarchical routing protocols*, also referred to as clustering protocols, superimpose a structure in the network, i.e., they give some nodes a special role in the communication process. Clustering was introduced in 80's to provide distributed control in mobile radio networks [97]. Inside the cluster one device is in charge of coordinating the cluster activities (*cluster head*). Beyond the cluster head, inside the cluster, we have: *ordinary nodes* that have direct access only to this one cluster head, and *gateways*, i.e., nodes that can hear two or more cluster heads [97]. As all nodes in the cluster can hear the cluster head, all inter-cluster communications occur in (at most) two hops, while intra-cluster communication occurs through the gateway nodes. Ordinary nodes send the packets to their cluster head that either distributes the packets inside the cluster, or (if the destination is outside the cluster) forwards them to a gateway node to be delivered to the other clusters. Only gateways and cluster heads participate in the propagation of routing control/update messages. In dense networks this significantly reduces the routing overhead, thus solving scalability problems for routing algorithms in large ad hoc networks. In traditional wireless scenarios the main goals of clustering are scalability and efficiency. In wireless sensor networks clustering is also used for data aggregation and energy-aware communication.

Several clustering algorithms have been proposed for wireless sensor networks (see [98] and [86] for additional information). One of the most popular is the Low Energy Adaptive Clustering Hierarchy (LEACH) [66]. LEACH divides network operations in two steps: a setup phase and a steady phase. In the setup phase cluster heads are selected by means of a random distributed algorithm. The non cluster-head nodes join the cluster which minimize the energy needed for communications. After the association procedure cluster heads create a cluster-wide schedule. The actual communication takes place during the steady phase. Sensing nodes collect data and transmit them to the cluster head. The cluster head performs aggregation and forwards the results to the sink. The steady phase is much longer than the setup phase to reduce protocol overhead. Moreover, the setup phase repeats periodically to ensure cluster head rotation.

In [99] the authors present a protocol called PEGASIS which improves LEACH by using a chain-based scheme. At first chains are constructed by using a greedy algorithm. Then data is transferred and aggregated along the chain. Only one node in the chain, i.e. the leader, transmits data to the base station. Leaders take turns to save energy when transferring data to the base station.

TEEN [100] and APTEEN [101] are threshold-based clustering protocols targeted to time critical applications, such as event detection. In TEEN cluster heads advertise two parameters, a hard threshold and a soft threshold. Nodes continuously sample the environment, but transmit to cluster heads only if the data is greater than the hard threshold. This limits energy consumption because the radio transceiver is kept in sleep mode for most of the time. In order to further reduce power, subsequent transmissions are allowed only if the variation of sensed data is greater than the soft threshold. Cluster heads periodically rotate in this case as well. APTEEN is an extension to TEEN in order to achieve better flexibility. APTEEN can dynamically change the operating parameters to match the application needs. In addition, APTEEN allows greater energy savings by means of transmission scheduling and aggregation.

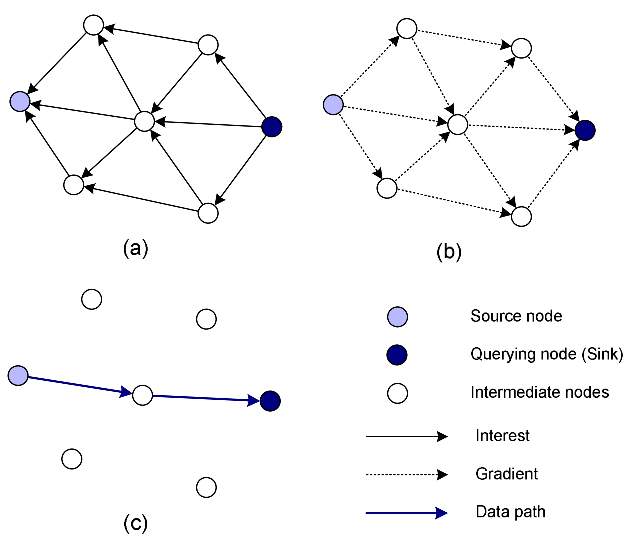


FIG. 17: Directed diffusion communication paradigm.

*Flat routing protocols* assume all nodes in the network behave the same for data processing and delivery, in contrast with the hierarchical approach. Flat routing follows the data-centric communication paradigm, i.e. in sensor networks data are more important than the individual nodes' identities. Thus, routing and forwarding inside a sensor network require a form of data-centric data dissemination to/from the sensor nodes. In this case, information is referred by using attributes of the phenomenon. For example, the query "tell me the temperature in the region *X*" needs to be disseminated to sensor nodes of a region *X*. At the same time, data coming from the region X have to be delivered to the user(s) issuing the query. Simple techniques such as flooding and gossiping can be used to disseminate the data inside the sensor network [87], [86]. However, these techniques waste energy resources by sending redundant information throughout the network. Several application-aware algorithms have been devised to efficiently disseminate information in a wireless sensor network. These algorithms are based on the publish/subscribe paradigm. Nodes publish the available data that are then delivered only to nodes requesting them. Dissemination algorithms achieve additional energy savings through in-network data processing based on data aggregation.

One of the most popular approaches is Directed Diffusion [102]. In directed diffusion each data is referred by an attribute-value pair. The sink broadcasts an interest that is a task description, containing a timestamp and a gradient (FIG. 17-a). The interest is linked to named data through the attribute-value pair. Each sensor stores the interest in a cache upon reception. Data dissemination, i.e. interest propagation, set up gradients related to data matching the interest (FIG. 17-b). When the originating node has matching data it sends through the interest gradient path (FIG. 17-c). Data propagation and aggregation are performed locally.

Directed diffusion inspired a number of similar protocols. For example, Gradient Based Routing (GBR) [103] improves directed diffusion using two different design choices. First, the interest includes a hop count (with respect to the sink), such that the gradient is set up along the minimum distance to the sink. Second, a number of data spreading and fusion schemes are employed to balance the load on sensor nodes, thus increasing the network lifetime. On the other side, Energy Aware Routing (EAR) [104] route data towards the sink along low-energy paths. To avoid depleting the energy of the nodes belonging to the minimum-energy path, EAR chooses one of multiple paths with a probability that increases the total network lifetime.

Similarly to directed diffusion, SPIN sends data only to sensor nodes which have requested them explicitly [105]. SPIN is based on a negotiation phase in which nodes exchange descriptors (i.e. metadata). Communications are more efficient because nodes send information describing the data instead of the data itself. First nodes advertise new data by using descriptors and wait for interested nodes to make request. The actual data is then transmitted. In addition, SPIN adapts the protocol behavior on the basis of nodes' remaining energy.

### C. Transport Layer

The sensor networks' data-centric nature combined with the strong resources' limitation make the Transport Control Protocol (TCP) protocol not suitable for the sensor network domain. Indeed, sensor networks require a sort of different concept of reliability. In addition, different reliability levels and/or different congestion control approaches may be required depending on the nature of the data to be delivered. The transport layer functionalities must be therefore designed in a power-aware fashion, to achieve the requested service level while minimizing the energy consumption at the same time. This implies using different policies for the forward path (from sensor nodes towards the sink) and the reverse path (from the sink towards sensor nodes).

In the forward path an event-reliability principle needs to be applied. The transport protocol does not have to correctly deliver all data. Instead, it must guarantee the correct delivery of a number of samples sufficient for correctly observing (at the user side) the monitored event. This can be done by exploiting spatial and temporal correlations between sensed data. Typically, sensor networks operate under light loads, but suddenly become active in response to a detected event and this may lead to congestion. In [106] an event-driven congestion control policy is designed to manage the congestion in the nodes-to-sink path by controlling the number of messages that notify a single event.

Indeed, the transport protocol should guarantee that, when an event is detected, the user correctly receives enough information. With ESRT [107] the concept of event-driven transport protocol introduced in [106] is extended to guarantee reliable event detection with minimum energy expenditure. The main operating parameters used by ESRT are the reliability observed by the sink and the reporting frequency. An analysis of the relations between these parameters leads to the definition of different operating conditions, each characterized by distinctive levels of reliability and congestion. The sink periodically broadcasts control packets with updated reporting rate in order to set the network in the optimal operating conditions.

The reverse path typically requires a very high reliability as data delivered towards the sensors contain critical information delivered by the sink to control the activities of sensor nodes (e.g., queries and commands or programming instructions). In this case more robust, and hence power-greedy policies must be applied, as proposed with PSFQ [108]. PSFQ slowly injects packets from sink to nodes by means of a controlled broadcast. This approach avoids interfering with the traffic coming from the other direction. On the other hand, PSFQ performs a more aggressive hop-by-hop packet recovery to overcome losses and out-of-order packets.

### D. Upper Layers

Sensor nodes in the sensing region *X* are typically set up to achieve in a cooperative way a pre-defined objective (e.g., monitoring the temperature in region *X*). This is achieved by distributing tasks to be performed on the sensor nodes. Therefore a sensor network is similar to a distributed system on which, at the same time, multiple applications are running. Each application is composed by several tasks that run on different (sensor) nodes. Starting from this view of a sensor network, in [109] the authors propose middleware-layer algorithms to manage, in a power-efficient way, a set of applications that may differ for the energy requirements and users' rewards. Specifically, the authors propose an admission control policy that, when an application starts, decides (given its energy costs and users' rewards) to accept/reject it to maximize the users' rewards. A policing mechanism is adopted, at runtime, to control that applications conform to the resource usage they declared at the admission stage.

The work in [109] is just an example of middleware-layer techniques for sensor networks. Due to space reasons we do not discuss the vast body of work in this area. The interested reader is referred to [110] for more details.


[1] V. Raghunathan, C. Schurghers, S. Park, M. Srivastava, "Energy-aware Wireless Microsensor Networks", *IEEE Signal Processing Magazine*, March 2002, pp. 40-50.

[2] G. Pottie, W. Kaiser, "Wireless Integrated Network Sensors, *Communication of ACM*, Vol. 43, N. 5, pp. 51-58, May 2000.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: a Survey", *Computer Networks* , Volume 38, N. 4, March 2002.

[4] *IEEE Pervasive Computing*, "Energy Harvesting and Conservation", Vol. 4, Issue 1, Jan-Mar. 2005.

[5] A. Kansal, J. Hsu, V. Raghunathan, J. Friedman, and M.B. Srivastava, UCLA Energy-Harvesting Project, http://nesl.ee.ucla.edu/projects/heliomote.

[6] L. Doherty, B.A. Warneke, B.E. Boser, K.S.J. Pister, "Energy and Performance Considerations for Smart Dust," *International Journal of Parallel Distributed Systems and Networks*, Volume 4, Number 3, 2001, pp. 121-133.

[7] J.A. Paradiso and T. Starnar, "Energy Scavenging for Mobile and Wireless Electronics", *IEEE Pervasive Computing*, Vol. 4, Issue 1, pp.18-26, Jan-Mar. 2005.

[8] J. Krikke, "Sunrise for Energy Harvesting Products", *IEEE Pervasive Computing*, Vol. 4, Issue 1, pp. 4-8, Jan-Mar. 2005.

[9] M. Philipose, J.R. Smith, B. Jiang, A. Mamishev, S. Roy, and K. Sundara-Rajan, "Battery-Free Wireless Identification and Sensing", *IEEE Pervasive Computing*, Vol. 4, Issue 1, pp. 37-45, Jan-Mar. 2005.

[10] A. Lal, R. Duggirala, and H. Li "Pervasive Power: A Radioisotope-Powered Piezoelectric Generator", *IEEE Pervasive Computing*, Vol. 4, Issue 1, pp. 53-61, Jan-Mar 2005.

[11] S. Roundy, E.S. Leland, J. Baker, E. Carleton, E. Reilly, E. Lai, B. Otis, J.M. Rabaey, P.K. Wright, and V. Sundararajan, "Improving Power Output for Vibration-Based Energy Scavengers", *IEEE Pervasive Computing*, Vol. 4, Issue 1, pp. 28-36, Jan-Mar 2005.

[12] X. Tang and J. Xu, "Extending Network Lifetime for Precision-Constrained Data Aggregation in Wireless Sensor Networks", *Proc. IEEE Infocom 2006*, Barcelona, Spain, April 2006.

[13] D. Ganesan, A. Cerpa, W. Ye, Y. Yu, J. Zhao, D. Estrin, "Networking Issues in Wireless Sensor Networks", *Journal of Parallel and Distributed Computing*, Vol. 64 (2004) , pp. 799-814.

[14] P. Santi, "Topology Control in Wireless Ad Hoc and Sensor Networks, *ACM Computing Survey*, Vol. 37, n. 2, p. 164-194, June 2005.

[15] Y. Xu, J. Heidemann, D. Estrin, "Geography-informed Energy Conservation for Ad Hoc", *Proc. ACM MobiCom 2001*, pp. 70 – 84. Rome, 2001.

[16] A. Cerpa, D. Estrin, "Ascent: Adaptive Self-Configuring Sensor Network Topologies, *Proc. IEEE INFOCOM 2002*.

[17] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris. "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks", *ACM Wireless Networks*, Vol. 8, N. 5, September 2002.

[18] F. Koushanfar, N. Taft and M. Potkonjak, "Sleeping Coordination for Comprehensive Sensing Using Isotonic Regression and Domatic Partitions", *Proc. of Infocom 2006*, Barcelona, Spain, April 2006.

[19] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring", *Proc. ACM Workshop on Wireless Sensor Networks and Applications*, pp. 88-97, Atlanta (USA), September 2002.

[20] X. Yang, N. Vaidya, "A Wakeup Scheme for Sensor Networks : Achieving Balance between Energy Saving and End-to-end Delay ", *Proc. IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2004)*, pp. 19-26, 2004.

[21] L. Gu, J. Stankovic, "Radio-Triggered Wake-up for Wireless Sensor Networks", *Real-Time Systems Journal*, Vol. 29, pp. 157-182, 2005.

[22] C. Schurgers, V. Tsiatsis, M. B. Srivastava, "STEM: Topology Management for Energy Efficient Sensor Networks", *IEEE Aerospace Conference '02*, Big Sky, MT, March 10-15, 2002.

[23] J. Rabaey, J. Ammer, T. Karalar, S. Li, B. Otis, M. Sheets, T. Tuan, "Picoradios for wireless sensor networks: the next challenge in ultra-low-power design", *Proc. International Solid-State Circuits Conference*, San Francisco, CA, February 3-7, 2002.

[24] L. Guo, J. Rabaey, "Low Power Distributed MAC for Ad Hoc Sensor Radio Networks", *Proc. IEEE Globecom 2001*, November 2001.

[25] M. Nosovich, T. Todd, "Low Power Rendezvous and RFID Wakeup for Embedded Wireless Networks", *Proc. of Annual IEEE Computer Communications Workshop (CCW 2000)*, Captiva Island (USA), October 15-18, 2000.

[26] E. Shih, P. Bahl, and M. Sinclair, "Wake on Wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices", *Proc. ACM MobiCom 2002*, Altanta (USA), September 2002.

[27] G. Anastasi, M. Conti, M. Di Francesco and A. Passarella, "An Adaptive and Low-latency Power Management Protocol for Wireless Sensor Networks", *Proc. of the 4-th ACM International Workshop on Mobility Management and Wireless Access (MobiWac 2006)*, Torremolinos (Spain), October 2, 2006.

[28] P. Buonadonna, D. Gay, J. Hellerstein, W. Hong, S. Madden, "TASK: Sensor Network in a Box", *Proc. European Workshop on Sensor Networks (EWSN 2005)*, January 2005.

[29] Q. Cao, T. Abdelzaher, T. He, J. Stankovic, "Toward Optimal Sleep Scheduling in Sensor Networks for Rare Event Detection", *Proc. International Conference on Information Processing in Sensor Networks (IPSN 2005)*, April 2005.

[30] B. Hohlt, L. Doherty, E. Brewer, "Flexible Power Scheduling for Sensor Networks", *Proc. ACM Workshop on Information Processing in Sensor Networks (ISPN 2004)*, Berkeley (USA), April 26-27, 2004.

[31] A. Keshavarzian, H. Lee, L. Venkatraman, "Wakeup Scheduling in Wireless Sensor Networks", *Proc. ACM MobiHoc 2006*, pp. 322-333, Florence (Italy), May 2006.

[32] Y. Li, W. Ye, J. Heidemann, "Energy and Latency Control, in Low Duty Cycle MAC Protocols", *Proc. IEEE Wireless Communication and Networking Conference*, New Orleans (USA), March 2005.

[33] G. Lu, B. Krishnamachari and C.S. Raghavendra, "An Adaptive Energy-efficient and Low-latency Mac for Data Gathering in Wireless Sensor Networks", *Proc. of 18th International Parallel and Distributed Processing Symposium*, Pages: 224, 26-30 April 2004.

[34] S. R. Madden, M. J. Franklin, J. M. Hellerstein and W. Hong, "TinyDB: an Acquisitional Query Processing System for Sensor Networks", *ACM Transactions on Database Systems*, Volume: 30, Issue: 1, Pages: 122-173, 2005.

[35] S. Madden, M. Franklin, J. Hellerstein and W. Hong, "TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks", *Proc. Annual Symposium on Operating Systems Design and Implementation (OSDI)*, 2002.

[36] D. Mirza, M. Owrang, C. Schurgers, "Energy-efficient Wakeup Scheduling for Maximizing Lifetime of IEEE 802.15.4 Networks", *Proc. International Conference on Wireless Internet (WICON'05)*, Budapest (Hungary), pp. 130 - 137, July 2005.

[37] T. Moscibroda, P. von Rickenbach and R. Wattenhofer, "Analyzing the Energy-Latency Trade-Off During the Deployment of Sensor Networks", *Proc. of Infocom 2006*, Barcelona, Spain, April 2006.

[38] N. Ramanathan, M. Yarvis, J. Chhabra, N. Kushalnagar, L. Krishnamurty, D. Estrin, "A Stream-Oriented Power Management Protocol for Low Duty Cycle Sensor Network Applications", *Proc. IEEE EmNetS II*, 2005.

[39] I. Solis, K. Obraczka, "The impact of timing in Data Aggregation for Sensor Networks", *Proc. IEEE International Conference on Communications (ICC 2004)*, June 2004.

[40] J. Elson, L. Girod and D. Estrin, "Time synchronization for wireless sensor networks", *Proc. IEEE International Parallel and Distributed Processing Symposium (IPDS 2001)*, 2001.

[41] S. Ganeriwal, R. Kumar, M. Srivastava, "Timing-sync Protocol for Sensor Networks", *Proc. ACM International Conference on Embedded Networked Sensor Systems (SenSys 2003)*, pp.138-149, 2003.

[42] V. Paruchuri, S. Basavaraju, R. Kannan, S. Iyengar, "Random Asynchronous Wakeup Protocol for Sensor Networks", *Proc. IEEE Int'l Conf. On Broadband Networks (BROADNETS 2004)*, 2004.

[43] J. Polastre, J. Hill and D. Culler, "Versatile Low Power Media Access for Sensor Networks", *Proc. of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November 3-5, 2004

[44] R. Zheng, J. Hou, L. Sha, "Asynchronous Wakeup for Ad Hoc Networks", Proc. *ACM MobiHoc 2003*, pp 35-45, Annapolis (USA), June 1-3, 2003.

[45] K. Arisha, M. Youssef, M. Younis "Energy-aware TDMA-based MAC for Sensor Networks", *Proc. IEEE Workshop on Integrated Management of Power Aware Communications, Computing and Networking (IMPACCT 2002)*, New York City (USA), May 2002.

[46] J. Li, G. Lazarou, "A Bit-map-assisted energy-efficient MAC Scheme for Wireless Sensor Networks", *Proc. International Symposium on Information Processing in Sensor Networks (IPSN 2004)*, pp. 56-60, Berkeley (USA), April 2004.

[47] V. Rajendran, K. Obracza, J. J. Garcia-Luna Aceves, "Energy-efficient, Collision-free Medium Access Control for Wireless Sensor Networks", *Proc. ACM SenSys 2003*, Los Angeles (USA), November 2003.

[48] I. Rhee, A. Warrier, M Aia, J. Min, "Z-MAC: a Hybrid MAC for Wireless Sensor Networks", *Proc. ACM SenSys 2005*, S Diego (USA), November 2005.

[49] G. Anastasi, E. Borgia, M. Conti, E. Gregori, A. Passarella, "Understanding the Real Behavior of 802.11 and Mote Ad hoc Networks, *Pervasive and Mobile Computing*, Vol. 1, N. 2, 2005.

[50] T.V. Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks", *The First ACM Conference on Embedded Networked Sensor Systems (Sensys'03)*, Los Angeles, CA, USA, November 2003.

[51] IEEE 802.15.4, *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, May 2003.

[52] W. Ye, J. Heidemann and 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.

[53] A. Ephremides O. Mowafi, "Analysis of a Hybrid Access Scheme for Buffered User Probabilistic Time Division", IEEE Transactions on Software Engineering, Vol. 8, N.1, pp. 52-61, January 1982.

[54] C. Schurgers, V. Tsiatsis, S. Ganeriwal, M. B. Srivastava, "Optimizing Sensor Networks in the Energy-Latency-Density Design Space", *IEEE Transactions on Mobile Computing*, Vol.1, No.1, pp. 70-80, January-March 2002.

[55] T. Armstrong, "Wake-up Based Power Management in Multi-hop Wireless Networks", available at http://www.eecg.toronto.edu/trevor/Wakeup/index.html.

[56] U. Malesci, S. Madden, "A Measurement-based Analysis of the Intercation between Network Layers in TinyOS", *Proc. European Workshop on Sensor Networks (EWSN 2006)*, LNCS 3868, pp. 292-309, 2006.

[57] G. Lu, N. Sadagopan, B. Krishnamachari, A. Goel, "Delay Efficient Sleep Scheduling in Wireless Sensor Networks", *Proc. IEEE Infocom 2005*, March 2005.

[58] Y. Tseng, C. Hsu, T. Hsieh, "Power Saving Protocols for IEEE 802.11 Ad Hoc Networks", *Proc. IEEE Infocom 2002*, New York (USA), June 2002.

[59] IEEE 802.11, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999.

[60] S. Jain, R. Shah, W. Brunette, G. Borriello, S. Roy, "Exploiting Mobility for Energy Efficient Data Collection in Wireless Sensor Networks", *ACM/Springer Mobile Networks and Applications*, Vol. 11, pp. 327-339, 2006.

[61] A. Kansal, A. Somasundara, D. Jea, M. Srivastava, D. Estrin, "Intelligent Fluid Infrastructure for Embedded Networks", *Proc. ACM SenSys*, pp. 111-124, Boston (USA), June 2004.

[62] P. Naik, K. Sivalingam, “A Survey of MAC Protocols for Wireless Sensor Networks”, Chapter 5 in *Wireless Sensor Networks* (C. Raghavendra, K. Sivalingam, T. Znati, Editors), Kluwer Academic Publishers, 2004.

[63] W. Ye, J. Heidemann, “Medium Access Control in Wireless Sensor Networks”, Chapter 4 in *Wireless Sensor Networks* (C. Raghavendra, K. Sivalingam, T. Znati, Editors), Kluwer Academic Publishers, 2004.

[64] I. Demirkol, C. Ersoy and F. Alagoz, “MAC Protocols for Wireless Sensor Networks: a Survey”, *IEEE Communications Magazine*, 2006.

[65] J. Haartsen, “The Bluetooth Radio System”, *IEEE Personal Communications*, Vo. 7, N. 1, pp. 28-36, February 2000.

[66] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, “Energy-efficient Communication Protocol for Wireless Microsensor Networks”, *Proc. Hawaii International Conference on System Sciences (HICSS-34),* January 2000.

[67] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D.E. Culler and K.S.J.Pister, “System Architecture Directions for Networked Sensors”, *Proc. of ASPLOS 2000*, November 2000.

[68] C. Buragohain, D. Agrawal, and S. Suri, “Power Aware Routing for Sensor Databases”, *Proc. of the 24$^{th}$ IEEE Conference on Computer Communications (Infocom 2005)*, Miami, FL, March 13-17, 2005.

[69] G. Xing, C. Lu, Y. Zhang, Q. Huang, R. Pless, “Minimum Power Configuration in Wireless Sensor Networks, *Proc. of the Sixth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2005)*, Urbana-Champaign, IL, May 25-28, 2005.

[70] R. Madan, S. Cui, S. Lall, and A. Goldsmith, “Cross-Layer Design for Lifetime Maximization in Interference-Limited Wireless Sensor Networks”, *Proc. IEEE INFOCOM 2005*, Miami (USA) March 13-17, 2005.

[71] Y. Shi, Y.T. Hou, H. D. Sheraliz, S.F. Midkiff, “Cross-Layer Optimization for Routing Data Traffic in UWB-based Sensor Networks”, *Proc. ACM MobiCom 2005*, Cologne (Germany) August 28 – September 2, 2005.

[72] S. Banerjee and A. Misra, “Minimum Energy Paths for Reliable Communication in Multi-hop Wireless Networks”, *Proc. ACM MobiHoc 2002*, Lausanne (Switzerland) June 9-11, 2002.

[73] H.S. Kim, T.F. Abdelzaher, W.H. Kwon, “Minimum-Energy Asynchronous Dissemination to Mobile Sinks in Wireless Sensor Networks”, *Proc. ACM SenSys 2003*, Los Angeles (USA), November 5-7, 2003.

[74] J. Luo, J.P. Hubaux, “Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks”, *Proc. IEEE Infocom 2005*, Miami (USA), March 13-17, 2005.

[75] W. Wang, V. Srinivasan, K.-C. Chua, “Using Mobile Relays to Prolong the Lifetime of Wireless Sensor Networks”, *Proc. ACM MobiCom 2005*, Cologne (Germany), August 28 - September 2, 2005.

[76] H. Yang, F. Ye and B. Sikdar, “SIMPLE: Using Swarm Intelligence Methodology to Design Data Acquisition Protocol in Sensor Networks with Mobile Sinks”, *Proc. of Infocom 2006*, Barcelona, Spain, April 2006.

[77] J.M. Reason, J.M. Rabaey, “A Study of Energy Consumption and Reliability in a Multi-Hop Sensor Network”, *ACM Mobile Computing and Communications Review*, Vol. 8, No. 1, pp. 84-97, 2004.

[78] S. Goel, A. Passarella and T. Imielinski, “Using buddies to live longer in a boring world”, *Proc. IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS 2006)*, Pisa (Italy), March 13, 2006.

[79] V. Kawadia, P.R. Kumar, “A Cautionary Perspective on Cross-Layer Desing”, *IEEE Wireless Communications*, pp. 3-11, Feb. 2005.

[80] D. Culler , P. Dutta , C.T. Ee , R. Fonseca, J. Hui, P. Levis, J. Polastre, S. Shenker, I. Stoica, G. Tolle, J. Zhao, “Towards a Sensor Network Architecture: Lowering the Waistline”, *Proc. USENIX International Workshop on Hot Topics in Operating Systems (HotOS 2005),* Santa Fe (USA), June 12-15, 2005.

[81] J. Polastre, J. Hui, P. Levis, J. Zhao, D. Culler, S. Shenker, I. Stoica, “A Unifying Link Abstraction for Wireless Sensor Networks”, *Proc. ACM SenSys 2003*, San Diego (USA), November 2-4, 2005.

[82] MobileMAN Project, Deliverable D13 “MobileMAN domain modeling”, available at `http://cnd.iit.cnr.it/mobileMAN/deliverables/mobileMAN-D13.pdf`, June 2005.

[83] M. Perillo and W. Heinzelman, “Wireless Sensor Network Protocols”, in *Algorithms and Protocols for Wireless and Mobile Networks* (A. Boukerche editor), CRC Hall Publishers, 2006.

[84] H. Karl, A. Willig, “A short survey of wireless sensor networks”, *TKN Technical Report* TKN-03-018.

[85] M. Vuran, I. Akyildiz, “Correlation-based Collaborative Medium Access in Wireless Sensor Networks”, *IEEE/ACM Transactions on Networking*, Vol. 14, N. 2, pp. 316- 329, April 2006

[86] K. Akkaya and M. Younis, “A survey on routing protocols for wireless sensor networks”, *Ad Hoc Networks*, Vol. 3, N. 3, May 2005, Pages 325-349

[87] J. Al-Karaki, A. Kamal, “Routing Techniques in Wireless Sensor Networks: a Survey”, *IEEE Wireless Communications*, Vol. 11, N. 6, pp. 6- 28, December 2004.

[88] J.-H. Chang and L. Tassiulas, “Maximum Lifetime Routing in Wireless Sensor Networks”, *Proc. of the*

*Advanced Telecommunications and Information Distribution Research Program (ATIRP'2000)*, College Park, MD, March 2000.

[89] K. Kalpakis, K. Dasgupta and P. Namjoshi, "Maximum lifetime data gathering and aggregation in wireless sensor networks", *Proc. of IEEE International Conference on Networking (NETWORKS '02)*, Atlanta, GA, August 2002.

[90] K. Sohrabi, J. Pottie, "Protocols for self-organization of a wireless sensor network", *IEEE Personal Communications*, Volume 7, Issue 5, pp 16-27, 2000.

[91] T. He, J. A Stankovic, C. Lu, T. Abdelzaher, "SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks", *Proc. of the 23rd IEEE International Conference on Distributed Computing Systems* (ICDCS'03), Providence, RI, May 2003.

[92] Q. Cao, T. He, L. Fang, T. Abdelzaher, J. Stankovic and S. Son, "Efficiency Centric Communication Model for Wireless Sensor Networks", *Proc. of Infocom 2006*, Barcelona, Spain, April 2006.

[93] S. Olariu and I. Stojmenovic, "Optimal Power Allocation in Wireless Networks with Transmitter-Receiver Power Tradeoffs", *Proc. of Infocom 2006*, Barcelona, Spain, April 2006.

[94] Y. Yu, D. Estrin, and R. Govindan, "Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks", *UCLA Computer Science Department Technical Report*, UCLA-CSD TR-01-0023, May 2001.

[95] V. Rodoplu and T.H. Ming, "Minimum energy mobile wireless networks", *IEEE Journal of Selected Areas in Communications*, Vol. 17, No. 8, pp. 1333-1344, 1999.

[96] L. Li and J. Y. Halpern, "Minimum-energy mobile wireless networks revisited", *IEEE International Conference on Communications (ICC'01)*, Helsinki, Finland, June 2001.

[97] D. J. Baker, A. Ephremides, and J. A. Flynn, "The Design and Simulation of a Mobile Radio Network with Distributed Control", *IEEE Journal on Selected Areas in Communications*, Vol. 2, N. 1, January 1984.

[98] S. Bandyopadhyay, E. J. Coyle, "An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks", *Proc. IEEE INFOCOM 2003*.

[99] S. Lindsey, C. Raghavendra, "PEGASIS: Power-Efficient Gathering in Sensor Information Systems", *IEEE Aerospace Conference Proceedings*, 2002, Vol. 3, 9-16 pp. 1125-1130.

[100] A. Manjeshwar, D. P. Agrawal, "TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks", *Proc. International Parallel and Distributed Processing Symposium (IPDPS'01) Workshops*, 2001.

[101] A. Manjeshwar and D. P. Agrawal, "APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks", *Proc. Int'l. Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, April 2002.

[102] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks", *Proc. ACM MobiCom 2000*, August 2000.

[103] C. Schurgers and M.B. Srivastava, "Energy efficient routing in wireless sensor networks", *MILCOM Proceedings on Communications for Network-Centric Operations: Creating the Information Force*, McLean, VA, 2001.

[104] C. Rahul, J. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks", *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, Orlando, FL, 2002.

[105] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks", *Proc. ACM/IEEE MobiCom '99*, pp. 174-85, Seattle (USA), August, 1999.

[106] C.-Y. Wan, , S.B. Eisenman, A.T. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks", *Proc. ACM SENSYS 2003*, November 2003.

[107] Y. Sankarasubramaniam, O. B. Akan, I. F. Akyildiz, "ESRT: event-to-sink reliable transport for wireless sensor networks", *Proc. ACM MobiHoc'03*, Annapolis (USA), June 2003.

[108] C. Wan, A. Campbell and L. Krishnamurthy, "PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks", *Proc. WSNA'02*, Atlanta (USA), September 2002.

[109] A. Boulis and M. Srivastava, "Node-Level Energy Management for Sensor Networks in the Presence of Multiple Applications", *Proc. IEEE International Conference on Pervasive Computing and Communications (PerCom 2003)*, Dallas (USA), March 2003.

[110] E. Yoneki and J. Bacon, "A Survey of Wireless Sensor Network Technologies: Research Trends and Middleware's Role", submitted for journal publication. Full version: *Technical Report UCAM-CL-TR646*, University of Cambridge, 2005, `http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-646.html`.