An Adaptive Low-latency Power Management Protocol for Wireless Sensor Networks

Giuseppe Anastasi•, Marco Conti*, Mario Di Francesco•, Andrea Passarella*

Pervasive Computing & Networking Lab. (PerLab)

•Department of Information Engineering
University of Pisa, Italy
{firstname.lastname}@iet.unipi.it

*CNR-IIT
National Research Council, Italy
{firstname.lastname}@iit.cnr.it

ABSTRACT

Energy conservation in wireless sensor networks is a critical issue. An efficient method to reduce power consumption consists in powering off the nodes' wireless transceiver when communication is not needed. Under this approach sleep/wakeup schedules of different nodes have to be synchronized. In addition, during the sleep phases nodes cannot communicate, and this might result in high delay. In this paper we introduce an adaptive and low latency power-management protocol based on sleep/wakeup schedules. The protocol is well suited for data collection applications in which sensors have to periodically report to a sink. It staggers the schedules of the nodes, in order to minimize the delay. One major advantage of this protocol is that the schedules are automatically adapted based on the network congestion and on the application traffic demand, so that the network can operate efficiently and completely unattended even in very dynamic conditions. Simulation results show that our power-management protocol effectively reacts to traffic and topology variations, without scarifying performance in terms of energy consumption, delivery ratio and delay. Furthermore it achieves lower energy consumption, collision ratio and delay than commonly adopted fixed sleep/wakeup schemes.

Categories and Subject Descriptors

C.2 [**Computer Communication Networks**]: Network Architecture and Design, Network Protocols, Network Operations, Distributed Systems

General Terms

Algorithms, Design, Management, Performance

Keywords

Wireless Sensor Networks, Power Management, Network Adaptation, Sleep Scheduling

1. INTRODUCTION

The increasing miniaturization of electronic components and the advances in wireless technologies has fostered research on wireless sensor networks. In one of the classic models, a sensor network consists of a large number of tiny sensor nodes deployed over a geographical area. Each node is a low-power device that integrates computing, wireless communication, and sensing capabilities. Sensor nodes are thus able to sense physical environmental information (e.g., temperature, humidity, vibrations, accelerations, and so on) and process the acquired data locally, or send them to one or more collection points (usually referred to as sinks or base stations) typically through wireless communications [4].

The key concern in wireless sensor networks is energy consumption, as sensor nodes are battery powered. The battery has limited capacity and often cannot be replaced nor recharged, due to environmental or cost constraints. Therefore, the design of a sensor network should be energy aware in order to prolong network lifetime. If we break down the energy expenditure of a sensor node we can see that typically the wireless radio consumes the highest energy share (much more than the sensing and processing components). For example, in a typical sensor node the energy cost of transmitting a bit of information is approximately the same as the cost of executing a thousand operations [5]. Furthermore, the power consumed when the radio transceiver is idle is often nearly the same as the power consumed in the transmit or receive mode [5]. Since the traffic load of typical applications is in the order of few tens of Kbps, energy consumption during idle phases is seen as a main concern. Techniques based on in-network data aggregation (such as TAG [6]) reduce the amount of data to be carried to the sink, thus helping to conserve energy. But the most effective way to reduce energy consumption is powering off the transceiver when communication is not needed.

In this case sensor nodes alternate between sleep and wakeup periods, and neighboring nodes need to coordinate themselves by implementing a sleep/wake schedule in order to make communication possible. This technique is referred to as duty-cycling. Unfortunately, designing efficient duty-cycling schemes is not straightforward. First, duty cycling introduces additional delays in the packet delivery, as packets cannot be transmitted until destination nodes wake up. Latency requirements are highly dependent on the application. For example, object tracking or event detection require quick response to the observed phenomena, so high latencies are not feasible. Designing energy

efficient solutions which at the same time achieve low latency in packet delivery is thus a challenging task. Second, most duty-cycling schemes use fixed parameters. In this case the ratio between wakeup and sleep periods is defined before the deployment, and once chosen it cannot be changed. Fixed duty-cycling schemes require rather simple synchronization mechanisms, but the designer has to know the network topology and the traffic pattern a priori. This is not always possible, and thus adaptive duty-cycle schemes are required. In this case there is no parameter to fine tune during the initial network deployment, because the protocol dynamically adapts the sleep/wakeup periods to observed operating conditions.

In this paper we propose a new adaptive power management protocol based on sleep/wakeup schedules, which enables both low-power and low-latency communication in wireless sensor networks. The power management protocol is targeted to data collection applications (e.g. environmental monitoring [1], [2]), in which sensor nodes have to periodically report to a sink. It organizes nodes in a logical tree rooted at the sink, and staggers wakeup phases of nodes in adjacent levels in the tree so as to minimize the delay towards the sink. With respect to other similar approaches, we make in this paper two main contributions. Firstly, our power-management scheme is not tight to any particular MAC protocol, and just requires a standard CSMA/CA kind of service. Secondly, it is able to quickly adapt the sleep/wakeup schedule to the operating conditions (e.g., traffic demand, network congestion, etc.). Simulation results show that, thanks to its adaptability, it outperforms commonly adopted fixed schemes in terms of energy consumption, data-delivery ratio, and delay.

The following sections are organized as follows. Section 2 presents and compares the different approaches related to duty-cycling power-management schemes. Section 3 describes our power-management protocol and its adaptation mechanism. Section 4 and 5 outline the simulation environment and discuss the simulation results, respectively. Finally, conclusions are drawn in Section 6.

2. RELATED WORK

Energy management for wireless sensor networks has been extensively studied in the literature. Different approaches has also been proposed to prolong network lifetime. In the following we present some duty-cycling schemes, since this is the area more closely related to this paper.

A class of power aware protocols include a low duty-cycle scheme for energy conservation at the MAC layer. B-MAC (Berkeley MAC) [8] defines a duty-cycle through a channel sampling technique called Low Power Listening (LPL) and an asynchronous sleep/wakeup scheme based on packets with long preambles. S-MAC (Sensor-MAC) [7] uses sync packets to coordinate the sleep/wakeup periods of nodes in the network. Every node can specify its own schedule or follow the schedule of a neighbor. T-MAC (Timeout MAC) [9] is an enhancement of S-MAC designed for variable traffic load. T-MAC defines a timeout based activation period: if no event occurs after the activation period a node can go to sleep. DMAC [10] is an adaptive duty-cycle protocol optimized for data gathering trees in sensor networks. DMAC exploits the knowledge of the topology in order to stagger nodes' schedules according to their position in the routing tree. Finally, the IEEE 802.15.4 MAC protocol [3]

supports a beacon enabled mode based on a superframe structure bounded by special synchronization frames called beacons. Each superframe is formed by an active period, where communication takes place, and an inactive period, in which nodes can sleep.

Another class of power management protocols is represented by general sleep/wakeup schemes. They do not define a medium access mechanism nor a way to manage the topology of the network, but they only specify how nodes synchronize with each other and rely on a lower MAC for actual data communication. Flexible Power Scheduling (FPS) [14] is a power management protocol specifically designed for data collection. FPS uses a coarse-grain time division in slots and a distributed algorithm for slot reservation. A node can be in a wake or sleep state during each slot and communicate exploiting an underlying MAC protocol. Schedules are local and dynamically adapt with respect to network demand. Finally, [15] presents an adaptive power conservation scheme targeted to data propagation. This scheme does not require the exchange of control messages and supports heterogeneous sensor nodes.

Duty-cycle based MAC protocols provide significant energy savings. However, they suffer from additional latency in packet forwarding. This *sleep latency* is introduced because a forwarding node has to wait until the destination wakes up before transmitting a packet, and increases with the number of hops. The most effective way to overcome this problem consists in exploiting upper layer information as in [10]. Another problem is that many duty-cycle schemes are fixed, in the sense that they cannot dynamically change their operating parameters, i.e. the length of the sleep/wakeup periods [8], [7]. In this case the correct choice of the duty cycle is an issue.

General sleep/wakeup schemes have major advantages over the other class. In fact, they are more flexible because they do not tightly depend on specific MAC and physical layers. At the same time, they can easily exploit additional network information provided by other protocols to achieve higher energy savings. FPS [14] has the major advantages of being adaptive and MAC independent. Unfortunately, FPS does not exploit the network topology to efficiently organize the sleep/wake period of nodes, therefore it can lead to high latencies. The scheme proposed in [15] is interesting, but seems suitable to very low data rate sensor network with non periodic traffic generation.

Our power management protocol belongs to the general sleep/wakeup schemes, thus it is independent of the MAC protocol used. In addition, it dynamically adapts to network traffic and exploits network layer information in order to minimize packet latency. Finally, it avoids using multiple radios to coordinate neighboring nodes.

3. PROTOCOL DESCRIPTION

3.1 Overview

In the following description we will refer to a data collection paradigm where data typically flow from the source nodes to a sink (data flowing from the sink to the sources are much less frequent). In addition, we will assume that nodes are organized to form a logical tree rooted at the sink used for data forwarding (*routing tree* or *data gathering tree*). Each parent node has to be physical neighbor of all its children. The data gathering tree may change over time due to route changes (e.g., caused by node failures). Also, it may be re-computed periodically to better share
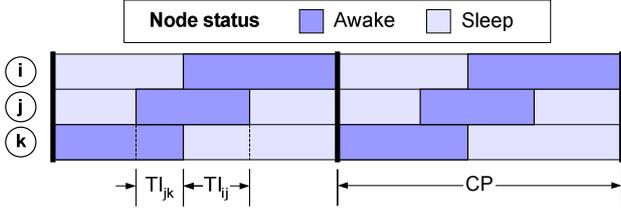
Figure 2. Sleep scheduling protocol parameters

(a) Talk interval reduction

(b) Talk interval increase

Figure 1. Talk interval variation examples

power consumption among nodes, thus prolonging the network lifetime. However, as nodes are assumed to be static, the data gathering tree – once established – remains stable for a reasonable time interval.

The basic idea behind our proposal is that in a typical data collection paradigm nodes can achieve low energy consumption and low latency in transferring data to the sink node if their active periods are staggered according to their position along the data gathering tree. Ideally, the active part should be the minimum amount of time to allow each node to receive data from its children, and send data to its parent node. In addition, the active part should vary dynamically in order to cope with variations in the traffic pattern, network congestion or topology.

In our scheme, communication between a parent and its children occurs in *communication periods* (CPs) that repeat periodically. Each communication period is divided into two portions: a *talk interval* (TI), during which nodes communicate by using the underlying MAC protocol, and a *silence interval* during which nodes are sleeping (Figure 2). The talk interval between a node and its children is adjacent to the one between the node itself and its parent in order to reduce the energy dissipation due to state transitions. Consider a generic node j having node i as parent and node k as child. Let CP^m denote the m-th communication period, TI^m_{ij} the talk interval between nodes i and j, TI^m_{jk} the talk interval between nodes j and k. Obviously, the following condition must hold to ensure the protocol correctness:

$$TI^m_{ij} + TI^m_{jk} \le CP^m$$

Information about the communication period and talk interval are advertised by parent nodes to children by periodically sending out special packets named *beacons*. Each beacon includes the time instant at which the next talk interval will start, and the duration of the next talk interval. Therefore, children know when they have to be awake to meet with the parent. Note that the protocol does not require precise synchronization among nodes. Simple guard-band mechanisms are included to avoid missing packets. Parent nodes send out a beacon at the end of each talk interval. In order to reduce the probability of collision with other packets, beacons are transmitted after a random delay within a beacon period, i.e. a reserved time period at the end of the talk interval.

As the talk interval adaptation policy is the core of our scheme, in the following subsections we will describe the algorithm used to estimate talk interval duration, as well as the protocol operations for varying the talk interval.

3.2 Adaptation algorithm
In our scheme the sleep/wakeup schedule is defined by the communication period and by the talk interval.
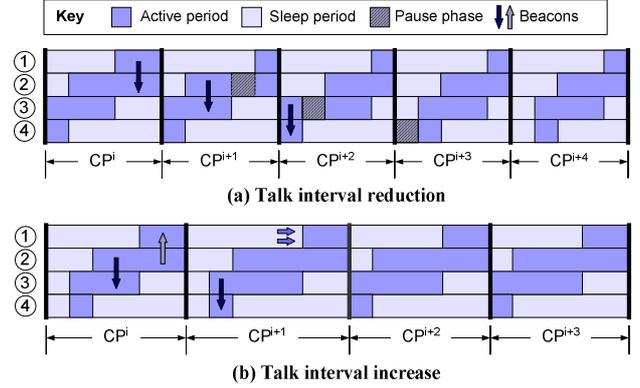
The length of the communication period is specified by the sink. In fact the communication period is closely related to the data collection interval. For example the user can query the maximum value of temperature, to be collected every 2 minutes. In this case the communication period is set to 2 minutes. A variation in the communication period corresponds to a modification of the query, i.e. the new interval for the periodic data acquisition.

Choosing an appropriate talk interval is somewhat more involved. The talk interval of a parent node should be tailored to the time needed to successfully receive all packets sent by its children. This time period depends on two factors: the number of packets and the MAC protocol used. The number of packets depends on the number of children, the packet generation at source nodes, and the topology. Note that in this work we do not assume any form of in-network aggregation, which is clearly a worst-case condition for our protocol. Thus, in a tree-based routing scheme, such as the one assumed in our approach, each non-leaf node forwards all packets coming from its descendants. On the other hand, the MAC layer used affects the time needed to complete a successful packet transmission. In fact the MAC protocol determines the channel access time, i.e. the time needed to acquire the channel before transmitting a packet.

The ideal value for each talk interval is the minimum value which allows the parent to successfully receive all packets coming from its children. From the above discussion it is clear that computing the ideal talk-interval value would require global knowledge of the topology. Moreover, this information would require to be updated as topology and network operating conditions change. Therefore, such an ideal approach is not feasible in a scenario when the number of nodes can be pretty high, and network conditions may change dynamically after the initial deployment.

To tackle these problems we propose an adaptive estimation technique that approximates the ideal scheme. Our approach lets every parent node choose its own talk interval. The decision involves only local information, thus it does not require global knowledge of the topology. In addition, the proposed technique leverages estimates of the channel access time for contention-based MAC protocols.

3.2.1 Talk interval estimation
In the following discussion we assume that nodes sample the environment before the beginning of their talk intervals with the parent. Therefore, at the beginning of a talk interval a child node

has all the packets it is going to send to the parent in a local buffer.

During a generic communication period, the parent node measures the following quantities:

- *Packet inter-arrival time* (Δ_i). The packet inter-arrival time is the difference between the time instants at which two subsequent packets are received. Δ_i s are stored for all received packets.

- *Number of received packets* (n_{pkt}). The total number of packets received in a single communication period.

These parameters refer to a single communication period., i.e. the current one. To smooth possible spikes in the estimator statistics, the estimate for the next communication period is computed using the values of the L previous communication periods. For this purpose each node uses two moving windows, in which it stores Δ_i and n_{pkt} related to the last L communication periods.

The time required to get all packets sent by children in the next communication period is then estimated as:

$$TI_{est}^{m+1} = \overline{\Delta} \cdot n_{pkt}^{max},$$

where $\overline{\Delta}$ is the average inter arrival time and n_{pkt}^{max} the maximum number of received packets (both statistics are evaluated on the respective moving windows). Note that using n_{pkt}^{max} is a conservative choice, to minimize the packet-loss probability. The estimate for the next talk interval is computed based on TI_{est}^{m+1}, and by recalling that the talk interval must also allow the parent node to send a beacon packet. Thus, the next talk-interval estimation is computed as

$$TI_{rnd}^{m+1} = \left\lceil (TI_{est}^{m+1} + BP)/s \right\rceil \cdot s.$$

The above equation needs some detailed explanation. First of all, we want the talk interval estimate to also include a beacon period, denoted by BP. Then, we introduce in the above equation the "slot-time", denoted by s. Specifically, the slot time is twice the maximum time required for a packet to be successfully delivered by the underlying MAC protocol. The talk interval estimates are an integer number of slots, and cannot be lower than one slot. This guarantees that any child has always a chance to send packets to the parent, even after phases during which it has no traffic to send.

Directly advertising TI_{rnd}^{m+1} to children as the next talk interval length might lead to some flapping of the protocol parameters. Therefore, in order to smooth the variation of the talk interval estimates, we also define two guard bands g_1 and g_2. The next talk interval advertised to children (TI^{m+1}) is finally computed as follows:

```
if ( TI_{rnd}^{m+1} − TI^{m} > g_1 )
        TI^{m+1} = TI_{rnd}^{m+1}
else if ( TI^{m} − TI_{rnd}^{m+1} ≥ g_2 )
        TI^{m+1} = TI_{rnd}^{m+1} − s
```

Increases of the talk interval are managed less conservatively than decreases. Indeed, if the difference between the new and the old estimate is higher than g_1, we immediately advertise the newly estimated value. On the other hand, if the difference between the old and the new estimates is greater than or equal to g_2 we
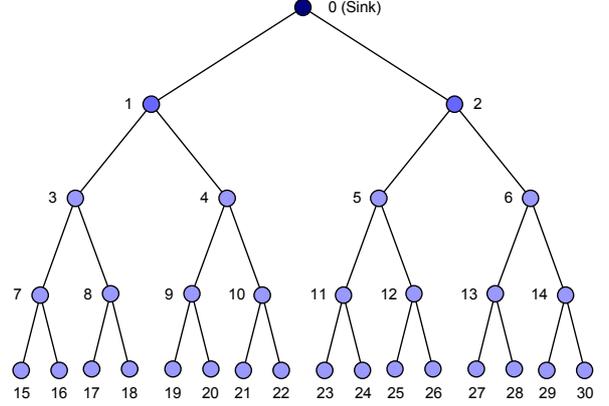
Figure 3. Balanced topology

decrease the estimate by one slot time. Indeed, aggressive increases tends to minimize the probability of loosing packets. Following the same rationale, g_1 should be less than g_2. In our simulations we have set $g_2 = 2g_1 = 2s$.

3.3 Protocol operations

Having laid down in the previous section the algorithm to adapt our scheme to dynamic network and application conditions, we now describe how this adaptation policy can be implemented into networking protocols.

3.3.1 Talk interval reduction

To reduce the talk interval a parent node advertises the new schedule parameters by sending a beacon to its children. All other nodes will automatically shift the communications ahead in time to ensure that their talk intervals are adjacent. To better understand how the protocol works, let's make reference to the scenario depicted in Figure 1a, and suppose that during the *i-th* communication period node 1 has decided to reduce its forthcoming talk intervals with its child (node 2). In the *i-th* communication period node 1 announces the new talk-interval duration to its child by means of a beacon. Node 2 receives the beacon and waits for the next communication period (*i+1*) to inform its child (i.e., node 3) about the new scheduling parameters. Because of this, node 2 introduces a *pause phase* between its talk intervals (one with its parent and another with its child). This behavior ensures that node 3 does not lose synchronization with its parent. The above actions are repeated by node 3 and its descendants (if any) in the next communication periods. Therefore, the pause phase shifts to lower levels one communication period at a time. A new steady-state schedule is reached after a number of communication periods equal to the depth of the routing tree.

In conclusion, the talk interval reduction shifts the communications to the right (with respect to time axis), deferring descendants sleep-awake schedules. This shift is an optimization to avoid idle times during wakeup phases. Note that nodes for which the talk interval is going to be reduced (nodes 1 and 2 in the figure) benefit from this shift already after one communication period. Also, handling the shift does not significantly increase energy consumption, since nodes can go to sleep during the pause phase (if it is long enough to make it convenient to switch off and on again).
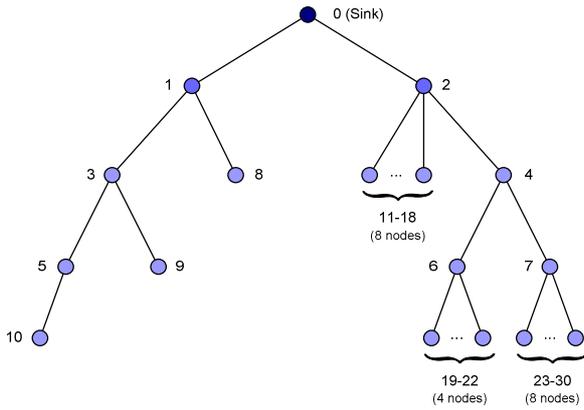
Figure 5. Unbalanced topology

3.3.2 Talk interval increase

To increase the talk interval duration, the parent node sends out two beacon packets. The first is sent to its children and contains the new schedule parameters. The second is sent upwards in the tree to force a communication shift ahead in time. This packet is required to ensure the correctness of the protocol, in order to achieve non overlapping parent-child schedules. As above, the example depicted in Figure 1b will help us understanding. Node 2 decides to increase its talk interval with its child, node 3. First, node 2 advertises the new talk interval value to its children. Second, in the same communication period, node 2 sends a special packet to its parent to shift its talk interval to the right.

The talk interval increase must satisfy some additional constraints. In fact it is easy to show that the talk interval increase of a parent must be less than the minimum sleep interval of its children.

4. SIMULATION ENVIRONMENT

In order to evaluate the proposed power management protocol we have implemented it using the ns2 simulator [11]. In all our tests we used the IEEE 802.15.4 in non-beacon enabled mode as the MAC protocol. We used the 2.4 GHz physical layer and turned on MAC layer acknowledgements. The radio propagation model was two-way ground; the transmission range was set to 15 m (according to the settings in [12]), while the carrier sense range was set to 30 m (according to the model presented in [13]).

4.1 Performance metrics

To evaluate the power management protocol we define the following performance metrics.

- *Delivery ratio*, as the ratio between the number of packets successfully received by the sink and the number of packets generated by source nodes.

- *Average latency*, as the time between a packet generation and the packet reception at the sink node.

- *Transient time*, as the number of communication periods that occur between a variation in the traffic and the stabilization of the new talk interval. We consider a talk interval as stable when it remains the same for at least *L* communication periods after the last change.

- *Power consumption*, since the main concern from an energy-consumption standpoint is reducing the duration of wakeup phases, we use the talk interval length also as
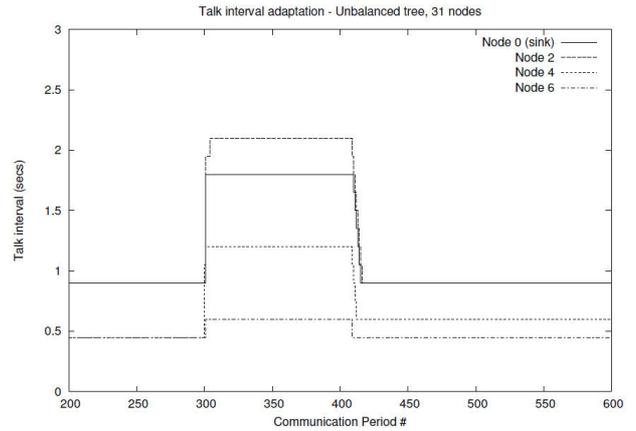
Figure 4. Talk interval adaptation for the traffic variation in the unbalanced scenario

power-consumption index.

Each experiment ran 10 times with independent replications. Unless otherwise stated, we report the average values, with 90% confidence intervals. The simulation duration was 1000 communication periods.

4.2 Topology configuration

We focused our analysis on two different scenarios. Both of them consisted of 30 nodes organized in a 4 level tree. The distance between each node and its parent/children was set to 7 m. The difference between the two scenarios lies in the way nodes were distributed along the tree.

- *Balanced topology*. The nodes formed a binary balanced tree rooted at sink. Leaf nodes laid only in the lowest level of the tree and each parent had exactly two children.

- *Unbalanced topology*. The nodes were unevenly distributed along the tree. There were leaf nodes at every level of the tree. In addition each parent could have a varying number of children.

The balanced topology (Figure 3) represents a scenario in which nodes can be carefully deployed, i.e. their position can be chosen by the designers of the sensor network. This topology also reflects the quite ideal situation in which the network conditions (number of children, traffic) are almost equally distributed along the tree.

On the other hand the unbalanced topology (Figure 5) corresponds to a scenario in which nodes are randomly deployed over a sensor field. In this case the designer has no control over the effective topology of the nodes. This can lead to very uneven distribution of the traffic in the network.

4.3 Dynamic conditions

To test the protocol adaptability we defined the simulation scenarios described below. In any case packets are generated only by leaf nodes. The other nodes only relay received packets to their parent. Therefore, performance figures presented hereafter are related to leaf nodes.

- *Traffic pattern variation*. In this experiment leaf nodes started generating one packet per communication period. From the 300th to the 400th communication period the traffic
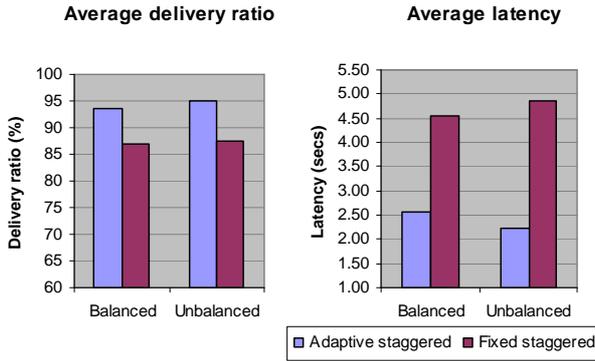
Figure 6. Global delivery ratio and latency for traffic variation

Figure 7. Delivery ratio distribution for traffic variation in the unbalanced scenario

tripled, i.e. each leaf node produced three packets per communication period. After the 400th communication period the nodes returned to the original generation rate of one packet per communication period.

- *Topology variation*. In this experiment only one half of leaf nodes were present in the initial topology. Starting from the 300th communication period and until the 400th, the remaining leaves joined the network. After the 400th communication period the topology returned to the original state. Each node produced one packet per communication period.

Table 1. Operational parameters for simulation

Parameter	Value
Talk interval slot (s)	150 ms
Beacon period	60 ms
Packet size (payload)	70 bytes
Communication period	30 s
Moving window size (L)	10

5. SIMULATION RESULTS

In the following subsections we show the simulation results obtained in the dynamic scenarios described before. In both cases, first we evaluate our power management protocol by analyzing its ability to adapt to network traffic. Then, we compare our approach and a staggered schedule with a fixed talk interval. Except where otherwise stated, the parameters used in the simulation are reported in Table 1.

5.1 Traffic pattern variation

5.1.1 Protocol adaptability

Figure 4 shows the variation in the talk interval of different nodes over the simulation time for the unbalanced topology during a representative simulation run. The talk interval value rapidly adapts to the traffic increase after the 300th communication period, then remains stable for about a hundred of communication periods and finally returns to the original value. A similar behavior is achieved with the balanced topology.

To better analyze the behavior of the power management protocol we focus on node 0 (the sink), which is influenced by the overall network traffic. The measures characterizing the transient of node 0 are shown in Table 2. The rising transient time is about five communication periods for the balanced topology and about four communication periods for the unbalanced one. This is related to data propagation. Talk interval adaptation is triggered by the additional packets generated by leaf nodes after the 300th communication period. To trigger the adaptation at the sink, such new packets have to "climb up" the tree, one level at communication period. Apart from the time required to trigger adaptation at the different levels, results show that the protocol reacts quickly to traffic increases. On the other hand, we can see that the falling transient time is longer, i.e. about 13 and 15 communication periods for balanced and unbalanced topology. This is the joint effect of two factors. Firstly, the system needs L communication periods to "forget" the previous high-traffic condition (in this case L =10). Recall that the talk interval is estimated based on the maximum number of packets a parent has received over the previous L communication periods. Secondly, recall that the adaptation heuristic we have adopted is quite conservative in reducing talk intervals, so as to minimize the probability of loosing packets. Clearly, this heuristic is a major driving factor of the performance in terms of raising and falling transient times.

Table 2. Sink adaptation transient times for traffic variation

Metric	Balanced	Unbalanced
Raising transient (CPs)	5.2 ± 2.9	3.6 ± 0.9
Falling transient (CPs)	13.4 ± 0.2	15 ± 0.01

It is also interesting to analyze the system performance in terms of delivery ratio and average delay. Specifically, in the first and last time slices – corresponding to the situation in which leaf nodes generate one packet per communication period – we get a delivery ratio (averaged over all leaf nodes) of about 97%, while the average latency experienced by leaf nodes is 1.8 seconds. During the time slice in which leaf nodes generate two packets per communication period, the delivery ratio drops to 92% while the latency increases to 2.6 seconds. This performance worsening is the effect of both the adaptability transient time, and the higher
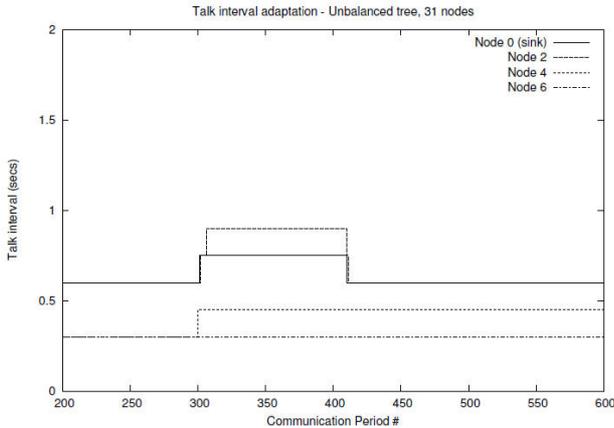
Figure 9. Talk interval adaptation for the topology variation in the unbalanced scenario

Figure 8. Global delivery ratio and latency for topology variation

contention in the network during this time slice. However, the delivery ratio and the delay still remain largely satisfactory.

5.1.2 Comparison with a fixed staggered approach

To have a comparison reference, we compare our adaptive staggered scheme against a popular fixed staggered one, like the scheme proposed for TAG [6]. In this case we set the value of the talk interval – that has to be the same for all nodes in the network – to the maximum value used by our protocol in the corresponding experiment, computed over all parent nodes in the network. This corresponds to taking the value that would fit the maximum traffic in the network during the entire experiment.

Figure 6 shows the overall latency and delivery ratio for the two approaches and the two scenarios. We can see that our adaptive protocol outperforms the fixed staggered scheme, both in terms of delivery ratio and delay. The higher delivery ratio – over 93% against about 87% – can be explained on the basis of contention. In our adaptive scheme brother nodes are free to choose their talk intervals (with their children) independently. On the other hand, in the fixed staggered approach it is easy to see that the starting time of the talk intervals of brother nodes will be always the same. If children of two brothers are able to interfere with each other at the MAC layer (which is quite likely), they will always contend (and possibly collide) with each other at the beginning of the talk intervals. In our adaptive scheme talk intervals defined by brother are independent of each other, and, as a side effect, this reduces the probability of collisions among children of brother nodes. Actually, we have found that the adaptive staggered scheme experiences about half the collisions experienced by the fixed staggered approach.

The performance gain of the adaptive scheme is even more evident if we focus on the delay and energy consumption indices. In terms of average delay, the adaptive staggered scheme obtains a latency of 2.56 and 2.22 seconds, respectively, for the balanced and the unbalanced topology. The fixed staggered scheme achieves 4.55 and 4.84 seconds average delays, instead, i.e., about twice the latency of the adaptive staggered scheme. This is due to the longer wakeup times used by the fixed scheme. Indeed, a node has to wait longer to relay data to its parent after having received them from the children. The higher wakeup time increases energy
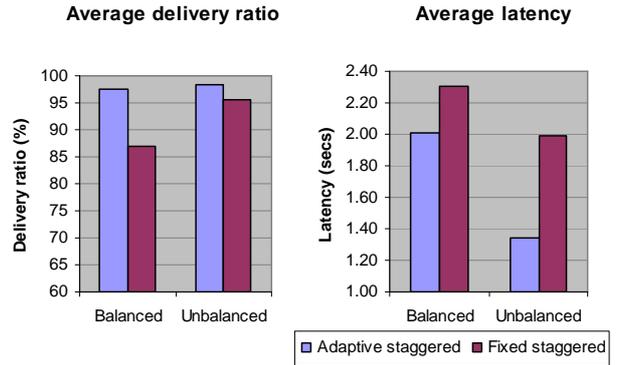
consumption, as well. In detail, the fixed staggered approach consumes about 2.5x the energy spent by the adaptive one.

To evaluate the fairness of the adaptive staggered approach with respect to the various leaf nodes, we also considered the delivery ratio distribution among leaf nodes in the network, as shown in Figure 7. In this case we consider the unbalanced scenario that is also the most critical one due to the way node are distributed along the tree. We can see that all nodes get almost equal delivery ratios, with minor variability with respect to the fixed staggered approach.

5.2 Topology variation

5.2.1 Protocol adaptability

In this experiment we analyze the experiments in which new leaf nodes were inserted after the 300th communication period, and then removed after the 400th communication period.

As in the previous case, Figure 9 shows the variation in the talk interval of different nodes over time for the unbalanced topology, during a representative simulation run. Again, the resulting behavior matches the expected variation. A similar behavior is achieved with the balanced topology.

Table 3. Sink adaptation transient times for topology variation

Metric	Balanced	Unbalanced
Raising transient (CPs)	1.8 ± 0.58	1.1 ± 0.42
Falling transient (CPs)	29.6 ± 7.6	10.0 ± 0.01

The measures characterizing the transient of the sink are shown in Table 3. We can see that the rising transient time is lower than in the previous experiment. This is due to the fact that in some cases the measured talk interval is already long enough to accommodate the increase due to the changed topology without increasing the talk interval. On the other hand, the falling transient time is higher than in the previous experiment. This depends on the lower variation of the traffic when the additional nodes are removed after the 400th communication period. Indeed, it can be shown that, under our policy, the talk interval decrease is quicker when the traffic variation is higher, because talk interval reductions are triggered more frequently.

Delivery ratio distribution
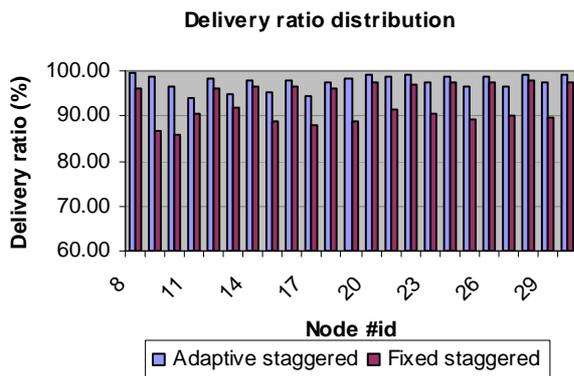
Figure 10. Delivery ratio distribution for the traffic variation in the unbalanced topology

5.2.2 Comparison with a fixed staggered approach

For comparison, we ran the same experiment with the fixed staggered scheme also in this case. Again, the adaptive staggered scheme achieves higher delivery ratios i.e., 94% and 95% against 87% and 88% for the balanced and the unbalanced topology, respectively. Specifically, also in this case the adaptive scheme halves the collisions experienced by the fixed staggered scheme.

The adaptive approach achieves performance gains also in terms of average delay and energy consumption. Specifically, the average delays under the adaptive scheme are 2.01 and 1.34 seconds for the balanced and unbalanced topologies. The corresponding average delay experienced under the fixed scheme are 2.30 seconds and 1.99 seconds. Finally, the fixed scheme spends about 1.4x the energy spent by the adaptive one.

As a final remark, Figure 10 shows that also in this set of experiments the adaptive scheme was able to be remarkably fair with all leaf nodes. The adaptive scheme outperforms the fixed scheme from this standpoint, as well.

6. CONCLUSIONS

In this paper we have proposed a staggered wakeup power-management protocol for sensor networks, suitable for data collection applications in which nodes have to periodically report to a sink. The main distinctive features of this scheme are: (i) it operates on top of the MAC protocol, and is not tight to any MAC in particular; and (ii) it is able to adapt the sleep/wakeup schedule dynamically, as the operating conditions change. These features are very well suited for sensor networks, which are expected to be deployed mostly at random, and have to run unattended after deployment. Simulations show that our power management protocol is able to rapidly adapt to varying traffic conditions. Specifically, it just needs a few application reporting periods to reach a new steady state after a sudden increase of either the traffic generated by the nodes, or the number of nodes in the network. In addition it also provides low latency data forwarding and low energy consumption. Compared to a fixed staggered approach, our scheme reduces latency and energy consumption by about one half. Furthermore, thanks to a drastic reduction of the contention, it also increases the delivery ratio.

The results presented in this work show that such an adaptive sleep/wakeup scheme is a valid candidate for sensor networks devoted to gathering data from the physical environment. We are currently working on better characterizing its behavior with respect to various networking scenarios it could be used in. Also, our scheme can accommodate various adaptation policies for dynamically changing the sleep/wakeup schedules. While in this paper we have presented a simple heuristic for this, different policies could be investigated and compared.

8. REFERENCES

[1] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler and J. Anderson, "*Wireless Sensor Networks for Habitat Monitoring*", Proceedings of the First ACM Workshop on Wireless Sensor Networks and Applications, Pages: 88-97 Atlanta, GA, USA, 28 September 2002

[2] G. Tolle et al., "*A Macroscope in the Redwoods*", 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys), San Diego, 2-4 November 2005

[3] 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

[4] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "*Wireless Sensor Networks: a Survey*", Computer Networks (Elsevier) Journal, Volume: 38, No: 4, March 2002

[5] V. Raghunathan, C. Schurgers, S. Park and M. B. Srivastava, "*Energy Aware Wireless Microsensor Networks*", IEEE Signal Processing Magazine, Volume: 19, No: 2, Pages: 40-50, March 2002

[6] S. Madden, M. Franklin, J. Hellerstein and W. Hong, "*TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks*", in Proceedings of OSDI, 2002

[7] W. Ye, J. Heidemann, and D. Estrin, "*An energy-efficient mac protocol for wireless sensor networks*", in Proceedings of the IEEE Infocom, New York, NY, June 2002, pp. 1567–1576

[8] J. Polastre, J. Hill and D. Culler, "*Versatile Low Power Media Access for Sensor Networks*", in Proceedings of SenSys '04, November , 2004

[9] T.V. Dam and K. Langendoen, "*An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks*", in Proceedings of Sensys'03, Los Angeles, CA, USA, November 2003

[10] G. Lu, B. Krishnamachari and C.S. Raghavendra, "*An Adaptive Energy-efficient and Low-latency Mac for Data Gathering in Wireless Sensor Networks*", in Proceedings of PDSP '04, Pages: 224, April 2004

[11] The Network Simulator Ns2, http://www.isu.edu/nsnam/ns

[12] J. Zheng and M. J. Lee, "*A Comprehensive Performance Study of IEEE 802.15.4*", IEEE Press Book, 2004

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

[14] B. Hohlt, L. Doherty and E. Brewer, "*Flexible Power Scheduling for Sensor Networks*", IEEE and ACM International Symposium on Information Processing in Sensor Networks, April 2004

[15] I. Chatzigiannakis, A. Kinalis and S. Nikoletseas, "*An Adaptive Power Conservation Scheme for Heterogeneous Wireless Sensors*", in the Proceedings of SPAA 2005, ACM Press, pp. 96-105, 2005