An Energy-Efficient Low-Latency Multi-sink MAC Protocol for Alarm-Driven Wireless Sensor Networks

António Grilo[1,2], Mário Macedo[1,3], and Mário Nunes[1,2]

[1] INESC, Rua Alves Redol, No 9,
1000-029 Lisboa, Portugal
[2] IST/UTL, Av. Rovisco Pais,
1096 Lisboa, Portugal
[3] FCT/UNL,
2825 Monte da Caparica, Portugal
{mario.nunes, antonio.grilo}@inesc.pt,
mmm@fct.unl.pt

Abstract. This paper presents a novel MAC protocol for Wireless Sensor Networks (WSN)s designated Tone-Propagated MAC (TP-MAC). This protocol is specially suited for early warning and tracking applications, where sensor nodes generate sporadic asynchronous traffic (mainly consisting of uplink alert messages and downlink control messages) with stringent latency requirements. This protocol aims to maximize energy-efficiency while minimizing latency in source-to-sink and sink-to-source communication. This difficult objective is achieved integrating scheduled channel polling (i.e. synchronized low power listening) with rapid fast path establishment based on the propagation of short wake-up tones. An analytical model was used to compare TP-MAC with SCP-MAC. The results show that TP-MAC is able to achieve better target latencies even when its duty-cycle is lower during periods of inactivity. The results also show that the advantage of using TP-MAC increases with the hop-distance between source and sink.

Keywords: Wireless Sensor Networks, Early Warning and Tracking, MAC, Energy-Efficiency, Scheduled Channel Polling.

1 Introduction

Wireless Sensor Networks (WSNs) have motivated intense research, in academia, industry and on the military sector due to the potential to support distributed micro-sensing in environments for which conventional networks are impractical or when the required sensor density demands a robust, secure and cost-effective solution. WSNs rely on large numbers of cheap devices, able to collaborate in distributed in-network data fusion and processing tasks, with final results that are equivalent to those obtained with centralized processing. An example of the latter is Homeland Security Early Warning and Tracking of Chemical, Biological Radiological, Nuclear and Explosive (CBRNE) agents, Toxic Industrial Materials (TIM), and other terrorist threats. In fact, this is generally regarded as one of the future WSN main applications.

J. García-Vidal, L. Cerdà-Alabern (Eds.): Wireless and Mobility, LNCS 4396, pp. 87 – 101, 2007.

Homeland Security Early Warning and Tracking is one of the WSN application scenarios addressed in the FP6 IST project Ubiquitous Sensing and Security in the European Homeland (UbiSeq&Sens). The overall objective of UbiSeq&Sens is to provide a comprehensive architecture for medium and large scale WSNs, with the full level of security required to make them trusted and secure for all applications.

Homeland Security Early Warning and Tracking poses interesting requirements on the WSN networking aspects, namely the requirements for low duty cycles (in order to assure maximum autonomy and minimum maintenance) and low latency in source-to-sink alert notifications (in order to assure a timely response to CBRNE/TIM threats). Current WSN MAC protocols usually trade-off one for the other, not supporting them simultaneously. The Tone-Propagated MAC (TP-MAC) protocol [1], presented in this paper, tackles this problem, supporting ultra-low duty cycles in periods of no activity, providing at the same time fast path establishment based on quick wake-up tone propagation in the beginning of activation periods. This paper extends the initial specification presenting a more comprehensive description, this time addressing synchronization and multi-sink support issues. This paper also proposes an analytical model for performance evaluation and discusses some obtained results.

The rest of this paper is organized as follows. Section 2 presents the WSN MAC protocols that were most relevant for this work. Section 3 presents TP-MAC. Section 4 presents an analytical model that will be used for the performance evaluation and comparison with SCP-MAC, made in section 5. Finally section 6 concludes the paper.

2 Related Work

Many techniques and MAC protocols have been developed for WSNs, in order to lessen the sources of inefficiency in wireless access, such as idle listening, collisions, message overhearing and control packets overhead [2]. We will focus on scheduled contention-based protocols, and low-power listening, as they are more related to our work.

Contention-based protocols are more flexible than Time-division multiple-access (TDMA) protocols, because they can provide more flexibility in multi-hop communications, and are more prone to topologic changes. The IEEE 802.11 standard [3] was designed for wireless LANs and also for ad hoc networks. In its Distributed Coordinated Function (DCF), it uses a contention protocol - Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), inherited from the MACAW protocol [4] – that has inspired some WSNs specific protocols, because it has satisfactory performance in avoiding collisions. Namely, the protocol messages sequence - RTS (Request to Send), CTS (Clear to Send), Data, and ACK (Acknowledge) – are used by other contention protocols.

S-MAC [5] is a scheduled contention-based protocol. Nodes are active (listening and transmitting) for some time, and asleep for the remaining time of a fixed period. In the active time, it uses the basic ideas of the IEEE 802.11/MACAW contention protocol for data exchange. A technique, named *virtual clustering*, permits that nodes adopt and propagate time schedules that synchronize the active time of the nodes in the sensor network. SYNC packets are exchanged to keep the nodes' schedules synchronized and compensate for clock drifts. However, *virtual clustering* leads to existence of multiple schedules [6], because they are generated locally by the initiative of the nodes. This is somehow a source of inefficiency, because nodes in the frontier of more than one schedule have more active times. In [6] is also presented an algorithm to achieve a unique global schedule, but convergence is very slow (in the order of several minutes). Other techniques are used by S-MAC in order to avoid overhearing, and excessive control packets overhead (such as message passing).

S-MAC duty cycle, and therefore its energy efficiency, can be tuned by acting on the active time duration. However, its fixed cycle of operation is not flexible to adapt to different load conditions. Some protocols addressed this problem, namely T-MAC [7].

The basic idea of T-MAC, which mainly differentiates it from S-MAC, is that active time duration is not fixed, but finishes when no activation events (i.e., communication activity) in the media occur for a chosen timeout. However, as its authors note, T-MAC suffers from the *early sleeping* problem: briefly, in a chain of four nodes, if the first node wants to communicate till the fourth, through the second and the third, the third node can still hear the first communication (between the first and the second), and remain active, but the fourth node goes too early to sleep state. This is a major drawback, because in WSNs the communication is mainly oriented to the sinks, in chain communication patterns. The authors propose two solutions for this problem, but the first solution (a Future Request to Send packet, issued by the third node) seems only to be effective to maintain the fourth node awoken, and not for subsequent nodes. Consequently, T-MAC seems to have a latency problem.

In order to reduce the latency, the research team of S-MAC also proposes a *fast path algorithm* [6], with additional and staggered active times along the path, between a node and the sink. However, the fast path must be reserved hop by hop, in the first packet communication. *Adaptive listen* [8] is another technique described in the context of S-MAC. Nodes that hear neighbors' protocol exchange messages, wake up for a short time, after the full data transmission ends, to check if they have transmissions for them, instead of going to sleep till the next active time. Analysis and simulations show the effectiveness of this solution.

The Data-gathering MAC (D-MAC) protocol presented in [9] includes an adaptive duty-cycle like T-MAC. However, its main purpose is to minimize the node-to-sink latency in convergecast[1] networks, where all sensing data converges to only one sink node. D-MAC uses staggered synchronization so that a data packet heard by a node at one level of the tree in one slot is transmitted to the next level in the following slot. The node is then allowed to sleep until the reception slot for its level occurs again.

Another approach, different from the scheduled schemes, is low-power listening (LPL) [10], used by B-MAC [11], and by WiseMAC [12]. It is a very simple mechanism designed to minimize the energy spent in idle listening: receiving nodes periodically poll the media for activity, and if there is no activity, they return to sleep state for the rest of the period; the sender node can wake up the receivers by sending a preamble. Poll durations can be very small, just the time to detect the preamble. However, preamble must last for an entire poll period, as nodes are not supposed to be synchronized. Nevertheless, this simple scheme can be effective in applications with low data traffic.

---

[1] Sometimes designated "reverse multicast".

More recently, a new scheduled contention protocol was proposed. SCP-MAC (Scheduled Channel Polling MAC, [13]) combines the advantages of LPL and scheduled protocols. Nodes that have data to transmit contend in a first contention window for tone transmission; nodes that win the contention transmit a tone. Possible collisions in tone transmission are allowed, because what is important is the presence of the tone. The potential receiving nodes poll the media for short time (around 2-3 ms), just enough to detect the tone. If there is no tone, the receiving nodes return to the sleep state. If there is a tone, they remain woken up for a further data transmission. Actual data transmission can be done with a second contention window, only with the winners of the first contention window, and with RTS-CTS exchanges.

Tone polls, and the tones themselves, are synchronized by the scheduled times, in the S-MAC way, and therefore can be very short. The long preambles of LPL are not needed. In this way, SCP-MAC proves to be much more efficient than LPL. Moreover, as contention is done in two consecutive windows, they can be smaller. Further use of adaptive listening in conjunction with SCP-MAC, is a hypothesis that the authors foresee. However, SCP-MAC presents a significant dependency between duty-cycle and transmission delay, since longer polling periods imply that the hop-by-hop transmission from sender to receiver also takes longer. This can be problematic in medium and large scale WSNs, specially in scenarios that require the combination of ultra-low duty cycles with low latency transmission (e.g. long-term deployment of alarm-driven applications), which is the main problem addressed by TP-MAC.

3 Tone-Propagated MAC (TP-MAC)

In order to achieve low duty cycle, the proposed TP-MAC protocol inherits some features from other MAC protocols, namely synchronized wake-up periods (S-MAC, SCP-MAC), and synchronized wake-up-tone announcement of data availability associated with scheduled channel polling (SCP-MAC). However, in TP-MAC the wake-up-tones are propagated across the WSN so that the nodes in the path from source to destination are woken-up as quickly as possible, before the arrival of the heralded data packets. In this way, TP-MAC is able to achieve low delivery latency even if the WSN node duty-cycle is extremely low, preventing or at least ameliorating the early-sleeping problem.

TP-MAC is based on the convergecast communication paradigm, assuming that the WSN is organized in a logical tree topology, associated with one sink, which corresponds to the root node. This imposes some cross-layer constraints on the network (i.e. routing) layer, which is not a real limitation, since most typical WSN scenarios require convergecast of sensor data towards sink nodes. Moreover, unlike D-MAC, TP-MAC supports the existence of multiple sinks in the network (and thus the coexistence of multiple overlaid logical trees – see below), but for sake of simplicity its basic mechanisms shall be first described assuming that there is a single sink node.

In a tree structure rooted at the sink node, it is possible to define different levels defined by the minimum hop distance relative to the sink node. In this way, the sink node constitutes level 0 and the level number increases as hop distance to the sink node increases. The establishment of network levels is at the core of the wake-up-tone propagation mechanism.

TP-MAC establishes super-frame periods for channel access, each starting by a synchronization wake-up-tone and two wake-up-tone propagation windows (upstream and downstream[2]), followed by a data transmission window (see Fig. 1). The size of the tone propagation window can be different for upstream and downstream, depending on the latency requirements. The channel access method in the transmission window can be based on any MAC protocol, e.g. CSMA/CA, S-MAC, T-MAC, SCP-MAC, etc.

The synchronization tone marks the beginning of the super-frame structure. This tone is periodically activated by the sink node and slowly propagated downstream to announce the transmission of a broadcast synchronizing/re-synchronizing SYNC packet in the data transmission window. The procedures supported by this synchronization tone are many, as shall be seen later.

The wake-up-tone propagation windows allow the announcement of data and establishment of fast paths from source to destination.

While no data traffic is generated, each node only has to poll the channel once in each wake-up-tone propagation window (only in the slot that corresponds to its level), and sometimes also in the synchronization slot. The nodes are allowed to sleep during the rest of the super-frame.

When a node has data to transmit, it first sends a wake-up upstream tone (e.g., for sensing data destined to the sink node), or a waking downstream tone (e.g., for control messages issued by the sink node to sensor nodes). The wake-up-tone propagation window structure guarantees that nearby nodes in the next upper/lower level listen to the generated wake-up-tone. They then propagate the tone upstream/downstream, as it can be seen in the tone propagation windows of Fig. 1. If a node detects a wake-up-tone in the last slot of a propagation window, then it shall only propagate it in the next super-frame. The tone propagation mechanism, which resembles the data propagation mechanism of D-MAC, assures that nodes within some hop distance are woken-up in just one operation cycle, forming a fast-path before actual data arrives. The maximum distance that a wake-up tone can reach in a single super-frame is equal to the number of tones in each tone propagation window, which is a configuration parameter.

The nodes that form a fast path stay active in the data transmission window, for a pre-defined time interval, which is dimensioned to keep those nodes active until the announced data arrives. The timeout mechanism is similar to that defined in T-MAC.

TP-MAC nodes only poll the media for a number slightly above two times per cycle (two polls, respectively for upstream and downstream propagated tones in each super-frame, and more seldom for the synchronization/re-synchronization tone), propagating the wake-up tones fast and deeply through the network (and thus opening fast data transmission paths). In this way it is possible to achieve low latencies simultaneously with low duty cycles.

One side effect of the TP-MAC protocol is that downlink propagation of wake up tones may result in waking up all nodes of the network. This behavior is an advantage when downstream data has to be broadcasted, and a handicap for other downstream

---

[2] In this paper, upstream and downstream definitions are relative to a WSN convergecast tree, considering the sink node at the top.

traffic patterns. Nevertheless, if the applications do not demand stringent latency requirements for downstream traffic, the downstream tone propagation window can be eliminated and the synchronization slot can be also used for data announcements. The frequency of the synchronization tone can be configured as to match the required latency. This solution decreases the duty cycle even further.

Another side effect is that for the upstream tone propagation the nodes in adjacent branches of the tree at levels above the source may be also woken-up, since they also detect and propagate the wake-up-tones. This behavior can cause some energy inefficiency. However, it is compensated by the ultra-low duty cycle required to achieve the intended delivery latencies.

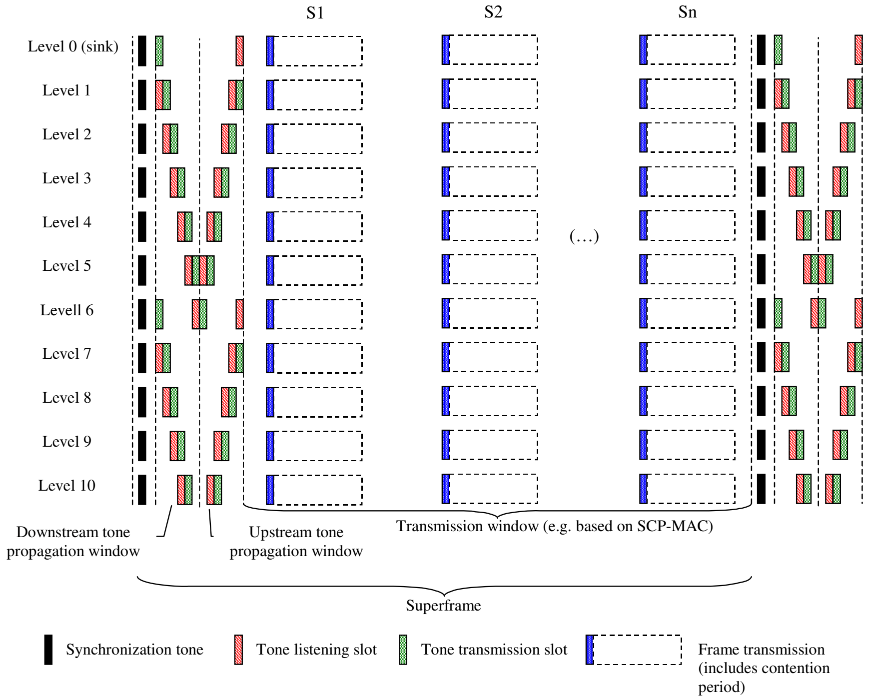

Fig. 1. TP-MAC super-frame structure and hierarchical wake-up-tone propagation

3.1 Synchronization / Re-synchronization

Like D-MAC, TP-MAC requires a global schedule spanning all levels of the logical tree. Being the root of the logical tree and natural destination of sensing data, the sink node is in a privileged position to initialize the WSN imposing the global schedule. Like in other scheduled MAC protocols, special SYNC packets are periodically issued in TP-MAC to re-synchronize the WSN and eliminate the effects of clock skew. In TP-MAC, these SYNC packets are firstly issued by the sink and propagated downstream in sequential super-frames. Each SYNC packet carries the sink node identifier and the current hop distance to the sink (which increases each time the

SYNC packet is propagated), allowing each receiving node to know its level in the tree. In order to announce the transmission of a SYNC packet in a super-frame, each issuing node precedes that super-frame by the transmission of a synchronization tone. Since the SYNC period is known by all nodes, the latter only have to listen to the synchronization tone in those specific super-frames where its transmission is expected. In case for some reason the synchronization tone is not heard, the node continues to poll the synchronization slot at the beginning of each subsequent super-frame until a synchronization tone is detected.

While this technique can be effective after the WSN is initiated, some difficulties arise just after deployment, when WSN nodes have no idea either about the global schedule or even about each other's schedules. A similar problem arises when new nodes are added to the network and must listen to a SYNC packet, before being admitted in the scheduled tree. One way to solve the problem is to force unsynchronized nodes to continuously listen to the channel until a SYNC packet is received. This procedure is energy consuming and risks draining the batteries, even before the network starts effective operation (especially when there is a large hiatus between WSN deployment and the initialization by the sink). The solution adopted in TP-MAC is to use LPL-based synchronization (see Fig. 2). SYNC packets are preceded by a long preamble, long enough to span the polling period of unsynchronized nodes (this period is a configuration parameter). Once the nodes detect any activity, they remain woken-up during the remaining of the preamble, in order to receive an ensuing SYNC packet. If the activity detected corresponded indeed to the preamble of a SYNC packet, those nodes get all the information needed to synchronize with the WSN. Otherwise, those nodes return to periodic polling the channel.

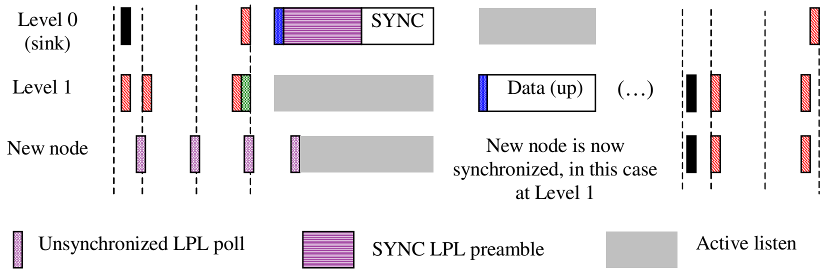

Fig. 2. TP-MAC LPL-based synchronization. The example considers a new node that is distant one hop from the sink node. Notice the long preamble of SYNC packets as compared to other packets.

3.2 Multiple Sink Nodes

Until now the description of TP-MAC only considered the existence of one sink node, located at the root of the tree. TP-MAC also supports the existence of multiple sink nodes, which is essential to satisfy the robustness and resiliency requirements of the UbiSeq&Sens Homeland Security Scenario. However, a single schedule is assumed, which means that the WSN must be fully initialized by an initial sink node, before additional sink nodes can join while adopting the initial schedule. The procedure to become a new sink is the same already described to initialize the WSN, i.e. the new sink activates the synchronization tone in the beginning of a superframe and in the subsequent transmission window it transmits a SYNC packet with its own identifier and hop distance to the sink equal to 0. This SYNC packet is propagated throughout the network in the usual way. Each receiving node creates an additional record for the new sink and adopts his new position in the tree in addition to the positions it already occupies relative to other existing sink nodes. This means that as the number of sinks increases, the number of times a node has to poll the channel during the tone propagation windows also tends to increase (there may be exceptions when a node is in such a position that it polls the same slot for two or more trees with different sinks). In the limit, a node may have to poll all slots in the wake-up tone propagation windows. The number of sinks should thus be limited in order not to affect energy-efficiency too much.

Since wake-up tones do not bear any information about the destination address, which may eventually be a multicast address, when a node belongs to two or more trees, each detected wake-up tone must be propagated in every other tree it belongs to, until the end of the current tone propagation window. While this technique presents the disadvantage of unnecessarily waking-up nodes in trees towards which the packet will not be sent, it keeps latency low since it avoids interrupting the propagation of

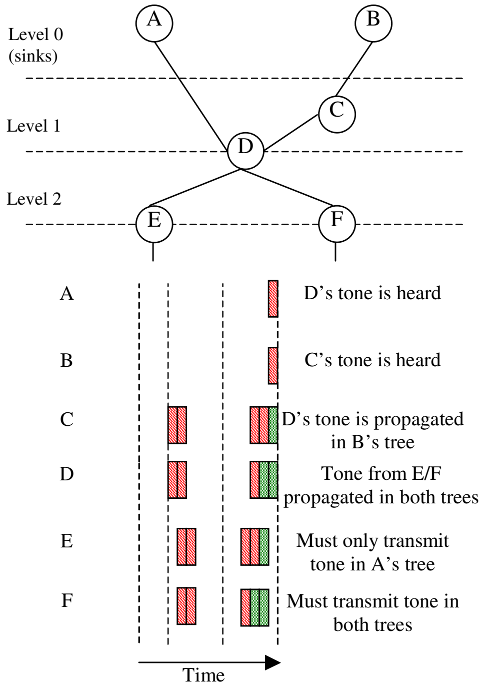

Fig. 3. Support of multiple sinks in TP-MAC

tones in any of the trees the node belongs to. This also solves a problem that arises when a node must poll the tone propagation window in two succeeding slots (one for each different tree to which the node belongs). Since the detection of a tone in the first of those slots would require it to be propagated in the ensuing slot, while the latter should also be polled, there would be a conflict if the detected tone was not propagated in every other tree. Fig. 3 illustrates this mechanism in a WSN comprising two sink nodes. In the example, node E generates a data packet directed to sink node A, while node F generates a data packet directed to sink node B. Since nodes E and F are in possession of the data packets, they know the respective destinations. Node E only transmits the uplink tone relative to level 2 (tree of sink node A), while node F must transmit two uplink tones because the tone relative to level 3 (tree of sink node B) falls in the slot that should be polled in level 2 (tree of sink node A). A similar situation happens at node D, which must transmit the received tone in both trees.

4 Analytical Model

In this section, we propose an analytical model for the performance evaluation of TP-MAC, based on a previous analytical model developed for SCP-MAC [13]. This model will also allow comparison of TP-MAC and SCP-MAC duty cycles for the same intended delivery latency.

Suppose that a data packet has to be routed to the sink node, and that the packet originator node is at distance of *N_hops* from the sink. This number is also the level of the node in the TP-MAC convergecast tree. We also admit that SCP-MAC delivers the packet through the minimum hop count path, and therefore this number is the same for a SCP-MAC network.

Considering that the packet is generated just after one SCP-MAC wake-up period, the worst-case delivery latency for SCP-MAC is approximately given by:

$$Td\_SCP \cong N\_hops \times Tp\_SCP \ , \tag{1}$$

where *Tp_SCP* is the cycle period of the SCP-MAC protocol. Expression (1) only applies if we assume, as it is the case for the sake of simplicity of the analysis, that there is just one single node that wants to communicate with the sink, and therefore there are no collisions with other transmissions. Moreover, in expression (1) we neglect the times of the SCP-MAC tone transmission, the time required for the second contention window and the packet transmission time at the last hop, because they are typically small. But, on the other hand, as we are about to equate expressions for delivery latency of the two protocols, in order to make a comparison of their duty cycles, all these terms cancel because they are equal in both protocols.

Delivery latency expression for TP-MAC is a little bit more complex.

The first term of the total latency is a TP-MAC cycle period, *Tp_TP*, which accounts for the worst case, when a packet is generated just after the tone propagation windows elapse. In terms of tone propagation and packet transmission, this first cycle is always lost.

Hereafter, we recommend the reader to follow the subsequent reasoning with the help of the Figure 1.

If the node level, *N_hops,* is lower or equal than the number of tones, *N_tones*, say for instance level 3, we can see that the tone is completely propagated to the sink in the propagation window of a second cycle. Subsequent data transmission can be done in a number of *N_hops* of consecutive data transmission slots of that cycle. The time required for these transmissions is equal to (*N_hops*-1)×*Tts_TP* (where *Tts_TP* is data transmission slot duration), plus the packet transmission time, and that of the SCP-MAC protocol overhead, the time durations of the tone propagation windows, and of the synchronization tone. We consider, in the following analysis, a time of *N_hops*×*Tts_Tp*, which is enough to account for those terms.

If *N_hops* is greater than *N_tones*, say for instance 8, another extra TP-MAC cycle, is needed. In the second cycle, propagation of the upstream tone and actual packet transmission are done only till level 6, because nodes in the levels above it, are still asleep (upstream tone could not reach them in the second cycle). In the third cycle, tone propagation is done till the sink, awaking a number of *N_tones* levels. Actual transmissions of the packet, in the third cycle, are done within a time equal to *N_Tones*×*Tts_TP* in that cycle.

It is interesting to note that the delivery latency in the upstream direction remains the same for nodes from level 7 till level 12. Propagation and transmission patterns only change in the second cycle, and remain the same in the third cycle.

When we increase the level from 12 (a multiple of *N_tones*) to level 13, another cycle is needed, and the number of cycles needed to transmit the packet is four.

With this reasoning we are able to derive the following expression:

$$Td\_TP \cong \left(\text{ceiling}\left(\frac{N\_hops}{N\_Tones}\right)\right)\times Tp\_TP + \\ \min\left(N\_hops, N\_tones\right)\times Tts\_TP \ .$$

Now it is worth to simplify this expression. If we approximate *Tts_TP* by the TP-MAC period, *Tp_TP*, divided by the number of slots of the data transmission window, *Nt_slots*, the above expression becomes:

$$Td\_TP \cong \left[\begin{matrix}\text{ceiling}\left(\dfrac{N\_hops}{N\_tones}\right)+ \\ \dfrac{\min\left(N\_hops, N\_tones\right)}{Nt\_slots}\end{matrix}\right]\times Tp\_TP \ . \tag{2}$$

Expressions (1) and (2) are of great utility, because they allow us to estimate the value of the TP-MAC period, needed to achieve the same expected latency as with the SCP-MAC protocol.

For SCP-MAC, we have one poll per cycle, and its poll frequency is given by the expression:

$$F\_poll\_SCP = \frac{1}{Tp\_SCP} \ . \tag{3}$$

Rigorous poll frequency estimation for TP-MAC, with data transmission in the upstream direction, is not an easy task, because as we have seen before, nearby nodes of those located in the minimum hop path, also form woken-up branches.

In order to compare the duty cycles of SCP-MAC and TP-MAC, we assume that the network remains silent for the most part of the time, and only sporadically has data to be delivered. We believe that this assumption is true for a large number of applications.

For TP-MAC, we have two polls per cycle, and more seldom one poll for the synchronization tone. If we define *Nc_st_TP*, as the number of TP-MAC cycles between two consecutive polls for the synchronization tone, the poll frequency for TP_MAC, for the quiet state of the network, is given by the expression:

$$F\_poll\_TP = \frac{\left(2 + \dfrac{1}{Nc\_st\_TP}\right)}{Tp\_TP} \quad . \tag{4}$$

The relation between the duty cycles of the two protocols is given by the following expression:

$$\frac{DC\_TP}{DC\_SCP} = \frac{F\_poll\_TP}{F\_poll\_SCP} \quad .$$

Using expressions (1) through (4), and equating the delivery latencies, the last expression becomes:

$$\begin{aligned} &\frac{DC\_TP}{DC\_SCP} \cong \left(2 + \frac{1}{Nc\_st\_TP}\right) \times \\ &\left[\text{ceiling}\left(\frac{N\_hops}{N\_tones}\right) + \frac{\min\left(N\_hops, N\_tones\right)}{Nt\_slots}\right] \times \quad . \\ &\frac{1}{N\_hops} \end{aligned} \tag{5}$$

Expression (5) can give the relation of the duty cycles of the two algorithms, in the quiet state of the network, needed to achieve the same target latency of a transmission, in the upstream direction, to the sink.

The relation of the two duty cycles decreases, as the number of hops increases. Therefore, TP-MAC becomes more energy efficient as the network size grows.

It is interesting to derive a limit for expression (5), as *N_hops* increases to high numbers. The ceiling term of expression (5) obeys the following inequalities:

$$\frac{N\_hops}{N\_tones} - 1 < \text{ceiling}\left(\frac{N\_hops}{N\_tones}\right) < \frac{N\_hops}{N\_tones} + 1 \quad .$$

Therefore the ceiling term, divided by *N_hops*, approaches $1/N\_tones$, as *N_hops* approaches infinity.

In expression (5), $\min(N\_hops, N\_tones) = N\_tones$ when $N\_hops > N\_tones$. Consequently, its division by *N_hops* approaches 0, as *N_hops* increases.

Therefore, the following expression can be written:

$$\lim_{N\_hops \to \infty} \frac{DC\_TP}{DC\_SCP} = \left(2 + \frac{1}{Nc\_st\_TP}\right) \times \frac{1}{N\_tones} \ . \qquad (6)$$

TP-MAP efficiency increases, as more tones are added, because more deeply and faster the wake-up tones are propagated to the sink, and data is transmitted. As the number of polls of TP-MAC remains the same for each cycle, lower latencies are achieved for the same duty cycle, or, conversely, lower duty cycles are obtained, for the same target latency. A large wake-up-tone propagation window becomes more attractive as the wake-up time increases, in order to minimize this overhead (typical wake-up times are in the order of 2 ms [14], which is approximately the expected duration of one wake-up-tone). However, it is not possible to increase indefinitely the number of tones, in order to achieve lower latencies, keeping the duty cycle unchanged, as this requires more data transmission slots, and greater tone propagation windows. On the order hand, if we increase the number of tones, in order to reduce the duty cycle keeping the latency unchanged, cycle periods become wider, demanding a decrease of the number of cycles between consecutive synchronization tone polls (i.e. *Nc_st_TP*), which has the opposite effect of raising the duty cycle.

5 Results

In this section some numerical results are displayed in two figures, in order to compare expected performance of the two MAC protocols.

Fig. 4 shows the poll period of the two protocols as a function of the target latencies, for a hop distance of 25 hops. The TP-MAC parameters are: number of tones: 6; number of transmission slots: 10; synchronization tone period: 5 cycles. The poll period of TP-MAC is slightly more than twice that of SCP-MAC, which means that, under the considered assumptions, it requires approximately one half of the SCP-MAC duty cycle, for the same target latency.

Fig. 5 shows the ratio between the duty cycles of TP-MAC and SCP-MAC as a percentage, for different numbers of hops, and different sizes of the wake-up tone propagation window. All other TP-MAC parameters are the same as those of Fig. 4. It is worth to note that TP-MAC duty cycle decreases with increasing number of hops, but that its energy efficiency gain, with respect to SCP-MAC, stabilizes for high numbers of hops. Higher number of tones can give higher energy efficiency gain. For instance, for 10 tones, we can obtain a duty cycle as low as 22% of the SCP-MAC duty cycle, for large network sizes.

These numbers seem to be very promising. We believe that TP-MAC can achieve very low duty cycles for the same target latency when compared with pure SCP-MAC, or equivalently very low latencies for the same duty cycles.

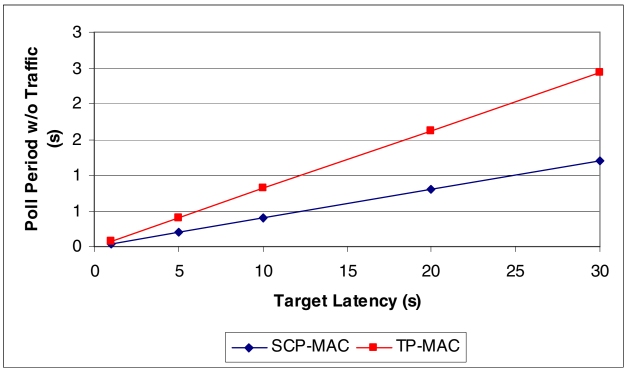

Fig. 4. Poll periods of TP-MAC and SCP-MAC as a function of the target latency

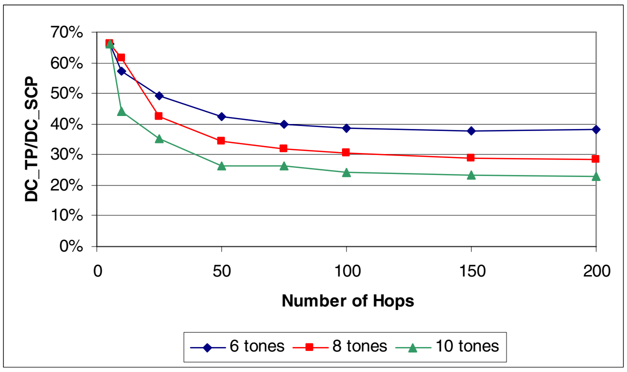

Fig. 5. Ratio between the duty cycles of TP-MAC and SCP-MAC as a function of the number of hops, and size of the wake-up tone propagation window

6 Conclusions

This paper has presented Tone-Propagated MAC (TP-MAC), a novel MAC protocol for WSNs, specially suited for early warning applications, where the traffic generated by sensor nodes (mainly alert messages) is sporadic, but has stringent latency requirements.

TP-MAC is based on a multi-sink multicast/convergecast tree WSN topology, supporting fast data path establishment, propagating short wake-up tones upstream and/or downstream between adjacent tree levels. This fast-path establishment mechanism does not incur on latency penalty for the transmission of the first packet of a stream, and it allows TP-MAC to achieve lower duty-cycles in periods of inactivity. The reliance on a tree topology is not a big limitation from the point of view of the authors, since it is typical in most WSN applications. The existence of a global schedule is essential to the operation of TP-MAC. The method used to establish and maintain this global schedule has also been addressed, as well as the mechanism for support of multiple sink nodes.

An analytical model to evaluate the performance (normal duty-cycle versus latency) of TP-MAC has also been presented, which allows a direct comparison with SCP-MAC. Performance results clearly show that TP-MAC can achieve much lower duty cycles, for the same target latency (e.g., 49% of the SCP-MAC duty-cycle for a 25-hop path between source and sink), or equivalently, very low latencies for the same duty cycles. This advantage tends to increase with the length of the path, and stabilizes for more than 50 hops. The analytical model was derived for the quiet state, as we believe that is the most common network state in many WSN applications. However, simulations and further development of the analytical model must be done, in order to compare TP-MAC and SCP-MAC duty cycles and latencies in other realistic scenarios, with increased contention and inter-flow interference.

Acknowledgments. The work described in this paper is based on results of IST FP6 project UbiSec&Sens. UbiSec&Sens receives research funding from the European Community's Sixth Framework Programme. Apart from this, the European Commission has no responsibility for the content of this paper.

The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

References

1. Grilo, A., Macedo, M., Nunes, M.: Tone-Propagated MAC (TP-MAC): A Low Duty-cycle Low Latency MAC Protocol for Wireless Sensor Networks. In: Proceedings of the Performance Control in Wireless Sensor Networks Workshop (hold in conjunction with the IFIP Networking Conference 2006). Coimbra, Portugal (2006).
2. Heidemann, J., Ye, W.: Energy Conservation in Sensor Networks at the Link and Network Layers. In: Bulusu, N., Jha, S. (eds.): Wireless Sensor Networks: A Systems Perspective. Artech House Inc. (2005) 75-86
3. IEEE: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std. 802.11 (1999).
4. Bharghavan, V., Demers, A., Shenker, S., Zhang, L.: MACAW: A media access protocol for wireless LAN’s. In: Proceedings of the ACM SIGCOMM. London, UK (1994) 212-225
5. Ye, W., Heidemann, J., Estrin, D.: An Energy-Efficient MAC protocol for Wireless Sensor Networks. In: Proceedings of the IEEE Infocom. New York (2002) 1567-1576
6. Li, Y., Ye, W., Heidemann, J.: Energy and Latency Control in Low Duty Cycle MAC Protocols. In: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2005). New Orleans, LA, USA (2005)
7. van Dam, T., Langendoem, K.: An adaptive energy-efficient MAC protocol for wireless sensor networks. In: 1st ACM Conference on Embedded Networked Systems (SenSys 2003). Los Angeles, USA (2003) 171-180
8. Ye, W., Heidemann, J., Estrin, D.: Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks. The IEEE/ACM Transactions on Networking, Vol. 12, No. 3. (2004) 493-506

9. Lu, G., Krishnamachari, B., Raghavendra, C.: An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks. In: Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS 2004). Santa Fe, NM, USA (2004)
10. Hill, J., Culler, D.: Mica: A wireless platform for deeply embedded networks. IEEE Micro, Vol. 22, No. 6. (2002) 12-25
11. Polastre, J., Hill, J., Culler, D.: Versatile Low Power Media Access for Wireless Sensor Networks. In: Proceedings of the 2nd ACM SenSys Conference. Baltimore, MD, USA (2004) 95-107
12. El-Hoiydi, A., Decotignie, J.-D.: WiseMAC: An Ultra Low Power MAC Protocol for the Downlink of Infrastructure Wireless Sensor Networks. In: Proceedings of the 9th IEEE International Symposium on Computers and Communications (ISCC'04). Alexandria, Egypt (2004) 244-251
13. Ye, W., Silva, F., Heidemann, J.: Ultra-Low Duty Cycle MAC with Scheduled Channel Polling. In: Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (SenSys 2006). Boulder, Colorado, USA (2006)
14. Körber, H., Wattar, H., Scholl, G., Heller, W.: Embedding a Microchip PIC18F452 based commercial platform into TinyOS. Workshop on Real-World Wireless Sensor Networks (REALWSN'05). Stockholm, Sweden (2005)