A Real-time MAC Protocol for Wireless Sensor Networks: Virtual TDMA for Sensors (VTS)

E. Egea-López, J. Vales-Alonso, A. S. Martínez-Sala, J. García-Haro, P. Pavón-Mariño, and M. V. Bueno-Delgado

Department of Information Technologies and Communications
Polytechnic University of Cartagena, Spain.
{esteban.egea, javier.vales, alejandros.martinez, joang.haro, pablo.pavon, mvictoria.bueno}@upct.es

Abstract. Wireless Sensor Networks (WSN) are designed for data gathering and processing, with particular requirements and constraints: low hardware complexity, low energy consumption, special traffic pattern support, scalability, and in some cases, real-time operation. In this paper we present the Virtual TDMA for Sensors (VTS) MAC protocol, which intends to support the previous features, focusing particularly on real-time operation. VTS adaptively creates a TDMA arrangement with a number of timeslots equal to the actual number of nodes in range. Thus, VTS achieves an optimal throughput performance compared to TDMA protocols with fixed size of frame. The TDMA frame is set up and maintained by a distributed procedure, which allows sensors to asynchronously join and leave the frame. A major advantage of VTS is that it guarantees a bounded latency, which allows soft real-time applications. An expression for the upper latency bound is also provided in this paper. VTS performance is evaluated by simulation. Results show less power consumption than other proposals in the field. We also introduce a novel multi-hop operation by coordinated sleep/awake cycles among clusters.

1 Introduction

Wireless Sensor Networks (WSNs) are a new paradigm of telecommunication networks. WSNs are designed to perform efficient data collection and environment monitoring, among other applications. WSNs share key properties with Mobile Ad-hoc NETworks (MANETs): decentralized control, wireless broadcast nature, self-configuring capabilities, multi-hop routing and ephemeral topologies. However, unlike MANETs, WSNs must support: (a) specific traffic patterns, characterized by very long idle periods and sudden peak transmissions, (b) long run battery-powered deployment, which yields to tight energy constraints, and (c) device (hardware and software) simplicity. Therefore, two fundamental goals of WSN protocols are energy saving and traffic/environment adaptivity. In addition, there are new incoming proposals of combined sensor and actors (devices that act upon events) networks [1], yielding the so-called Wireless Sensor and

Actor Network (WSAN) model. WSANs are by nature alarm-driven systems, where the reaction time (from sensor detection to actor action) must usually be bounded. Thus, WSAN proposals must add real-time operation as a requirement for their associated protocols.

WSNs major sources of energy waste are related to radio communication issues [2]. Namely, *collisions*, *idle listening*, *overhearing* packets addressed to other nodes, and *packet overhead* (sending and receving too many control packets). Since nodes do not know when they will receive packets from their neighbors, they are always listening to the channel (idle listening) and the radio is kept in receiving mode, consuming energy. Reference [3] states that idle listening is the dominant factor. Thus, radios must be *turned off* during periods of inactivity to save energy. Besides, sudden trafffic peaks are likely to happen in WSNs. High loads may collapse the network, degrading its performance (throughput and latency) and raising power consumption. Consequently, adaptation to extreme situations is mandatory for WSN protocols. Device limitations (both hardware and software), additionally impose that algorithms and protocols be simple.

In this paper we propose the VTS (Virtual TDMA for Sensors) MAC protocol. VTS provides a TDMA-like access scheme, in which the number of available slots *dynamically* adjusts to the number of nodes present in a cell (cluster) of nodes. Such a mechanism, after a transient adjustment phase, leads to a *scalable* and collision-free MAC protocol that consumes *considerably less energy* than contention-based protocols and has a *bounded packet latency* (providing support for soft real-time services). VTS also addresses network setup and synchronization issues. The trade-off is the average latency, which is slightly worse than contention protocols under low/medium loads.

As most of the sensor network proposals [2, 5, 8], VTS periodically puts nodes to sleep to reduce power consumption, which results in *listen/sleep* cycles. Our protocol employs a synchronization procedure similar to S-MAC [2] to establish the *listen/sleep* schedule. However, unlike S-MAC, only one node can transmit in every *listen/sleep* cycle. Thus, every cycle becomes what in a TDMA context is called a *timeslot*[1]. By following an extremely simple procedure, the nodes in a cluster will transmit in different timeslots. Therefore, when each node is finally transmitting in a different timeslot, a *frame* of timeslots has been built in a *distributed way*. VTS allows frame adjustment, that is, to increase or reduce the number of timeslots, which improves throughput compared to a TDMA frame with a fixed number of timeslots. With this TDMA-like access there is no contention for data transmission and latency is guaranteed.

Besides, in a multi-hop network VTS achieves good performance. Border nodes maintain time-shifted TDMA arrangements for each cell they belong to. This assumption implies that separate clusters of nodes independently select listen periods which do not overlap. It is a reasonable assumption because the usual listen interval only lasts around 10% of the cycle time. We call this operation mode Awake Time Division Multiple Access (ATDMA).

[1] In this paper, we refer to a listen/sleep cycle as timeslot, cycle or frame, depending on the context. A set of listen/sleep cycles is called a superframe.

The rest of the paper is organized as follows: Section 2 contains related work on MAC protocols for WSNs. In Section 3 the basis of the S-MAC protocol is reviewed to introduce VTS synchronization procedure. Section 4 thoroughly describes the VTS protocol. A performance analysis of VTS is presented in section 5. Finally, section 6 concludes and suggests future work.

2 Related Work

A considerable research effort has been devoted to WSNs in the last few years. Many new protocols and applications are currently being proposed and tested. WSN MAC protocols focus mainly on energy efficiency. Latency in message delivery is not usually a metric to be optimized. Most of the proposals can be classified in one of the classical categories: contention or TDMA-based.

MAC contention protocols are simple, scalable and flexible. Their major drawback is a high idle listening time (the dominant factor of energy waste). WSN contention-based proposals presently extend the Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) mechanism, applying additional schemes to reduce overhearing and idle listening: (1) Out-of-band signaling requires additional radio channels [4], and hardware is more complex and expensive. (2) Coordinated scheduling of listen time, which was first proposed by the Sensor-MAC (S-MAC) protocol [2]. S-MAC introduces a procedure to synchronize nodes on a common structure, that yields a shared listen/sleep cycle among neighbor nodes. This schedule reduces idle listening and, therefore, energy consumption. The Timeout-MAC (T-MAC) [5] protocol improves S-MAC by using an adaptive cycle length. The listen/sleep interval duration adapts to traffic fluctuations and obtains a better energy profile. This family of MAC protocols is relatively simple but does not guarantee latency. In contrast, with a similar complexity, *our protocol keeps latency bounded* (see section 4).

TDMA protocols assign timeslots to nodes, avoiding collisions and idle listening. However, in *ad-hoc* and sensor networks, establishing and maintaining a superframe of timeslots is a complex task. In addition, if the number of nodes dynamically changes, which is likely to occur in WSNs, scheduling must be readapted. *All* TDMA proposals for WSNs (and MANETs) utilize contention stages to setup and maintain a properly organized TDMA. Our protocol also belongs to this category. There is a number of these proposals for MANETs and WSNs: (1) The Five Phase Reservation Protocol (FPRF) [6], which provides a distributed algorithm to solve the problem of slot allocation in multi-hop networks. FPRF allocation procedure performs well at the expense of a great complexity, and does not implement any energy saving mechanism. (2) In Eyes MAC (EMAC) [7] a node can be active or passive. Active nodes own a timeslot and form a network backbone that performs routing tasks. Passive nodes use contention periods to send data. EMAC is focused on the increase of network lifetime, whereas latency or throughput are not addressed. (3) The Lightweight Medium Access Protocol (LMAC) protocol [8] is a modification of EMAC in which each node selects a timeslot using slot occupancy information from its one-hop neighbors. Its main limitations are that the number of available slots is fixed and the nodes listen to unused slots. Therefore, LMAC latency and throughput degrades at low loads [9]. In comparison with these proposals, VTS *is simpler and does not fix the number of timeslots*, therefore, it achieves a better performance.

3 S-MAC Overview

S-MAC [2] is a contention-based protocol that reduces energy consumption by means of several mechanisms.

Periodic listen and sleep forces nodes to activate periodically for a small time interval (the listen period); the rest of the time the nodes turn off their radio and sleep (the sleep period). A listen/sleep cycle is also called a frame (see Fig. 1). The ratio of the listen interval to the sleep interval is the *duty cycle*. Neighbors achieve and maintain a *coordinated sleeping* time, synchronizing their listen/sleep schedules by means of the short SYNChronization (SYNC) packet. SYNC packets correct clock drifts and are used to discover new neighbors. In a stationary situation, each node broadcasts a SYNC packet after a fixed number of frames (N_C) to maintain synchronization. Within a frame, the listen interval is subdivided into SYNC period (for SYNC packets) and Data period (for data packets), as shown in Fig. 1. Nodes perform carrier sense during a random number of slots (contention) before transmitting SYNC. If two nodes contend for transmitting a SYNC in the same cycle, it may happen that: (1) Nodes choose a different number of carrier sense slots. As a result, the node with a higher number defers its SYNC transmission (losing contention), and makes another attempt in the next synchronization cycle. (2) Both nodes select the same slot. A collision occurs which is not detected by any of them. After N_C cycles, both nodes contend again for SYNC transmission. Transmission of an information packet, occurs in the Data period inside the frame (see Fig. 1). Nodes can make use of the Data period in any frame: synchronization and Data periods operate independently.

Collision avoidance is based on CSMA/CA. It uses a RTS (Request To Send)-/CTS (Clear To Send)/Data/ACK sequence, with a fixed backoff contention window. Notice that S-MAC uses two independent periods of contention in every cycle, one for SYNC and one for Data transmission.

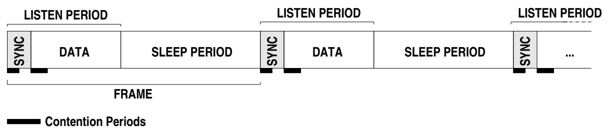


Fig. 1. S-MAC listen-sleep frame

To avoid *overhearing*, all the nodes sleep either at the beginning of the sleep period or inmediately after receiving a RTS or a CTS not addressed to them, and they wake up when the next frame starts. This scheme (periodic listen and sleep) significantly reduces idle listening. However, there is an undesirable effect on packet latency, because nodes must wait for the next listen period to send their data. To overcome this issue, S-MAC proposes a technique called *adaptive listening*: nodes which overhear a RTS or CTS packet wake up at the end of the transmission, instead of waiting for their next scheduled listen time. Thereby, if a node is the next-hop destination, its neighbor is able to inmediately pass the data to it.

Summarizing, S-MAC reduces idle listening and provides an efficient mechanism to synchronize nodes. Nevertheless, it does not act upon the other major sources of energy waste: collisions and packet overhead. It even increases them: there are two contention intervals (SYNC and Data) every cycle, and a new control packet (SYNC). Moreover, S-MAC cannot guarantee packet latency.

3.1 S-MAC Synchronization

In S-MAC, when a node initializes, it keeps listening for a certain amount of time. If it receives a SYNC packet, it adopts its listen/sleep schedule and tries to send its own SYNC in the next available chance. Otherwise, the node chooses its own schedule and broadcasts it using a SYNC packet. After N_C listen/sleep cycles, nodes broadcast a new SYNC packet to maintain synchronization.

Figure 2 illustrates this effect for a network with M neighbor nodes, being node A_1 the first node sending a SYNC. Any other node will follow A_1 schedule. The rest of the nodes compete to send a SYNC packet in the next scheduled SYNC time. Nodes that lose contention compete again every cycle until they send a SYNC. Let the i-*th* listen/sleep cycle be called *timeslot* i (t_i), the evolution of the network is as follows:

- At t_1, node A_1 sends the first SYNC. A cycle counter is set to N_C, which decreases by one every cycle. When the cycle counter reaches 0 a new SYNC is delivered.
- At t_2, the rest of the nodes try to send a SYNC packet, but only the contention winner sends it. Let us assume that an arbitrary node A_2 wins contention and so it sets its cycle counter to N_C. We say that node A_2 has *captured* this timeslot.
- At t_3, the remaining nodes try to transmit their SYNC packet, but, again, only the contention winner actually sends it. Node A_3 wins the contention (captures timeslot) and sets its cycle counter to N_C.
- At t_M, the last present node sends its SYNC packet and sets its cycle counter to N_C.
- From t_{M+1} to t_{N_C}, there are timeslots without SYNC transmissions.
- At t_{N_C}, A_1 sends a SYNC again.
- At t_{N_C+1}, A_2 sends a SYNC again.
- And so on.

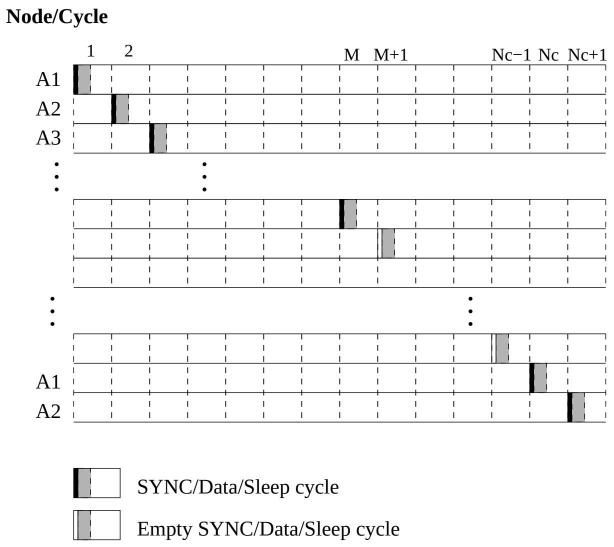


Fig. 2. S-MAC and VTS setup evolution

Let us notice that S-MAC implicitly defines a TDMA-like arrangement of N_C timeslots, even though, in fact, it is not used, because S-MAC allows *all* nodes to contend for sending data *every* listen period. On the contrary, *VTS takes advantage of this property to setup and maintain an adaptive TDMA frame.*

4 VTS Description

As stated in section 2, TDMA access schemes are the natural way to keep latency bounded and to reduce energy consumption, since there are neither contention nor collisions. VTS constructs a TDMA structure with the *exact number of timeslots needed*, that is, VTS dynamically adjusts the number of timeslots (N_C) in the TDMA frame to the number of nodes in the cell. In a stationary state, the protocol ensures that each node owns a single timeslot, not shared with any other node. In this situation, a *virtual superframe of timeslots* is created. The word *virtual* means that nodes do not know the superframe arrangement: neither its limits, nor their relative position in the superframe. They just *independently* keep a counter with the superframe length (N_C). VTS synchronization procedure works as S-MAC (Sect. 3.1), but, unlike S-MAC, VTS nodes are only allowed to send data in their captured cycle, i.e., *nodes only send packets, any kind of packet, every N_C cycles after their firstly sent SYNC packet.* It can be seen in Fig. 2 that a superframe of length N_C is virtually established.

Briefly, a VTS node contends every cycle until it captures a timeslot (wins the contention). From then on, the node only sends packets every N_C cycle. After a number of network setup (initialization) cycles, the nodes adjust their superframe length counter to their number of known neighbors. If nodes leave the cell, the superframe length is distributedly reduced. Finally, to allow new nodes to join the superframe, there is always a short contention period before packet transmission, where new nodes can contend with the owner of the timeslot.

In the following sections a detailed description of VTS operation is provided, starting with the single-hop network description and followed by the extension of VTS for multi-hop topologies.

4.1 Network Setup

The network setup phase lasts from nodes activation, to the definition of a (still of fixed size) virtual superframe. In VTS we prefer the name ConTroL (CTL) packet to SYNC packet, because the packet is used also for other purposes which are discussed in next sections. VTS setup stage behaves exactly as the S-MAC synchronization mechanism presented in section 3.1. That is, a node contends to send a CTL packet every listen cycle until it actually sends it. Then, the node only can send a new CTL packet after N_C cycles, and it *must* send it as a keep-alive beacon. When all the nodes have sent their first CTL packet, the virtual superframe of N_C timeslots is formed. Let us notice that nodes are not aware of the implicit timeslot allocation. They just know that they are only allowed to transmit packets every N_C cycle. Thus, the slot allocation procedure is simple and fully distributed. Let us also notice that nodes know who their neighbors are when they receive CTL packets. In the example of Fig. 2 it is assumed that all the nodes are initialized simultaneously and that they always capture timeslots consecutively. Let us see what would happen if these assumptions did not hold:

1. *If a node is initialized after the superframe has been formed.* It will wait for a CTL packet to join the listen/sleep schedule. Once it has been received, the new node tries to send its own CTL packet in the next scheduled timeslot. If this timeslot is owned by another node, both of them contend for the timeslot. The contention winner becomes the owner of the timeslot. The looser retries to send its CTL packet in the next timeslot. If this timeslot is also owned by a node, the contention winner will own the timeslot and the looser will keep trying. Eventually, an empty timeslot will be reached, which is captured by the only remaining node trying to access the medium at that moment. If more than one new node initialized, the only difference would be that the number of contending nodes for a timeslot would be higher. Eventually, every node would be assigned an empty timeslot. We call this process of multiple deallocation and reassignment “allocation loop” .
2. *If more than one node selects the same access instant during contention there is a collision.* In this case, two or more nodes send their CTL packets simultaneously. However, the contenders are not aware of the collision and each one considers that its own CTL has been correctly sent. After N_C cycles (the superframe length) they will try to send their CTL packet again, so there will be a new contention. The contention winner will own the timeslot. The looser will keep trying to send the packet, as discussed in the previous case. In the unlikely event of a new collision this sequence is repeated.

4.2 Adjustment of Virtual Superframe Length

With the previous setup procedure, a virtual superframe of fixed length N_C is created. If N_C were kept fix, protocol performance would be poor. On one hand, if N_C is less than the maximum number of neighbors, nodes cannot exclusively own a timeslot. In this case, there would always be contention between at least two nodes in every slot. On the other hand, if N_C is greater than the actual number of neighbors, VTS latency and throughput are negatively affected, since they are proportional to superframe length.

To overcome these situations, the number of timeslots should be adapted to the actual number of nodes. Therefore a node adjusts the initial cycle counter (set to N_C) to the real number of neighbors in the cell, that is, *to the number of received CTL packets from distinct nodes up to that moment.* This is done a number of timeslots (let it be N_S, a protocol parameter) after node initialization, From then on, the node dynamically adapts to the possibility of nodes joining and leaving the cell:

1. *New nodes join the cell*: anytime a node receives a CTL packet from an unknown neighbor, the superframe length is updated ($N_C = N_C + 1$).
2. *Nodes leave the cell*: within a superframe the mandatory CTL packet from the timeslot owner must be received. Therefore, CTL serves as a keep alive beacon, which allows to signal missing neighbors. However, a single CTL packet missed does not mean that its corresponding node has actually left the network: its CTL packet may have been corrupted or it may have been a collision with a new node joining the cell. Hence, a node is considered missed only after a certain number of inactivity timeslots (let it be N_I). On such event, the frame length and number of known nodes is updated ($N_C = N_C - 1$).

Let us note that in both cases there is a transient period before stability: in the first case the incoming node "steals" a slot, and it causes a new allocation loop. In the second case, nodes are not aware of the position of the lost node in the superframe, so they cannot properly adjust their cycle counter. To overcome it nodes randomly select a value within zero and the number of known neighbors.This solution requires a full reallocation of positions in the superframe. Since these events are supposed to be unlikely, this scheme was preferred because it keeps the protocol extremely simple. It should be remarked here that sensor nodes do not usually move, that is, networks are assumed to be static, and a node leaves the cell only when it has depleted its battery.

4.3 Data Exchange and Control Packets

At the beginning of each timeslot, all the nodes wake up and listen. The owner of the timeslot performs a carrier sense (choosing a random slot from a fixed contention window), and broadcasts a single and short control packet (CTL), see Fig. 3. which is used as:

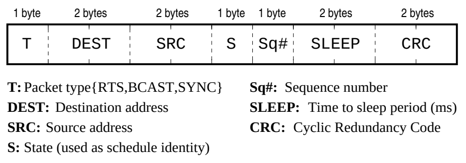

1 byte
2 bytes
2 bytes
1 byte
1 byte
2 bytes
2 bytes
T
DEST
SRC
S
Sq#
SLEEP
CRC
T: Packet type{RTS,BCAST,SYNC}
DEST: Destination address
SRC: Source address
S: State (used as schedule identity)
Sq#: Sequence number
SLEEP: Time to sleep period (ms)
CRC: Cyclic Redundancy Code

Fig. 3. CTL packet format

- Synchronization and schedule discovery (as S-MAC SYNC packet).
- Keep-alive beacon. It is mandatory for a node to send a CTL packet in its owned slot, since its neighbors must know that the node is active.
- New node discovery. CTL packets include source address. Thus, new nodes are added to the list of known neighbors as CTL packets arrive.
- Channel reservation: RTS information is included in CTL packets. This way, non-addressed nodes may go to sleep just after CTL packet reception

VTS uses the CSMA/CA mechanism for data delivery. The following types of transmissions exist:

1. Unicast packet transmission. A $\text{CTL}_{\{\text{RTS}\}}$ packet is sent by the owner of the timeslot. Non addressed nodes change to the sleep state inmediately, avoiding overhearing. Destination node replies using a CTS. Transmission is finished after a Data/ACK sequence and both nodes go to sleep.
2. Broadcast packet transmission. A $\text{CTL}_{\{\text{BCAST}\}}$ packet is sent by the owner of the timeslot. Destination is a broadcast address. All the nodes keep listening. Inmediately, sender sends the broadcast packet, that is, without waiting for any CTS reply. After receiving the packet nodes go to sleep. No ACK is sent.
3. No data transmission. A $\text{CTL}_{\{\text{SYNC}\}}$ packet is sent. Nodes adjust the clock reference, clear sender inactivity counter and go to sleep.

Control packet overhearing is reduced this way. A single CTL packet performs synchronization and discovery, reservation and keep-alive functions.

4.4 Single-hop Cluster Latency

Let T_C be timeslot duration. In a single-hop cluster in steady-state (i.e., all the nodes are the owners of a timeslot) any data transmission between two nodes has a maximum latency (L) given by:

$$L \leq N_C T_C \tag{1}$$

This is the maximum expected latency considering that MAC layer does not enqueue packets. Figure 4 illustrates this expression. Let us assume that the node A_1 generates a packet for any other node in cluster A. In the worst case, this packet arrives just at the end of the activity period (label T_0 in figure 4), so a superframe (of length $N_C T_C$) must pass before packet transmission ends (label T_1 in the figure).

4.5 Multi-hop Network

Large sensor networks are usually organized in *clusters* [10] , with the border nodes sharing coverage areas between adjacent clusters. Information coming from one cluster to another must hop among them. VTS proposes the Awake Time Division Multiple Access (ATDMA) scheme for multi-hop networks. That is, nodes coordinate awakening among clusters in order to have one-hop neighbor clusters never wake up at the same time, to avoid inter-cluster interferences. In other words, all the clusters except one are in their sleep period. When this active cluster goes to sleep, another one wakes up, and so on. Thus, a VTS border node forms different superframes with its neighbors in each cluster. The length of both superframes depends on the number of neighbors in each cluster and it may be different. Once superframes are created, packets can travel from one cluster to other during its corresponding listen interval. Hence, ATDMA satisfies cluster operation without interference. Combining VTS and inter-cluster ATDMA two goals are achieved: (1) Bounded multi-hop latency and (2) Spatial reuse of channel.

ATDMA is based on the assumption that clusters adopt different listen/sleep schedules, since they are independently initialized. Which means that virtual superframes in every cluster are time-shifted. In this case, border nodes see several schedules and adopt all of them. ATDMA operation is also depicted in the example of figure 4, where two neighborhoods share the media using ATDMA. Nodes in the first cluster are A_1, A_2, A_3 and A_4. Nodes in the second cluster are B_1 and B_2. To clarify ideas, let us assume that A_1 is a border node between the two clusters A and B, and therefore, it also owns a cycle in cluster B. Let us note that ATDMA seamlessly allows a different number of nodes in each cluster.

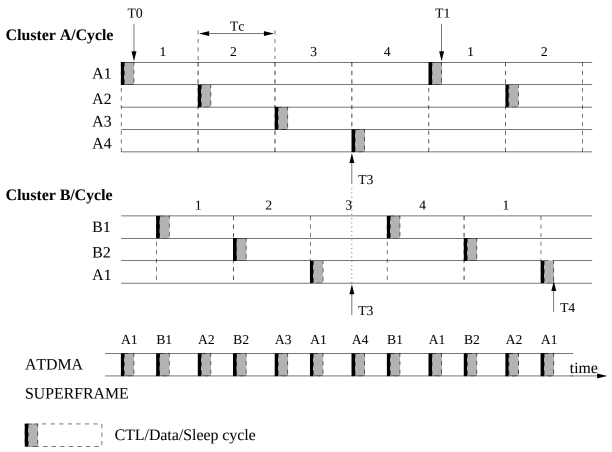


Fig. 4. ATDMA superframe evolution with two adjacent clusters (A and B) with 3 and 2 nodes respectively, plus one common border node A_1

However, if there exists overlapping two problems arise: (1) A border node cannot know in which schedule each of its neighbors is. VTS still works because there is always a channel sense period before transmission, but the latency *cannot be guaranteed.* (2) The border node will suffer from hidden node collisions. Nodes in adjacent cells will own the same slot in pairs, causing collisions at the boder node.

Therefore, VTS must ensure non overlapping schedules. In any case, ATDMA can be easily achieved after running a network-layer clustering protocol like [10], that can shift sleep time between superframes. At the moment, we assume that border nodes always see non-overlapping schedules. In a clustered network, with random initialization, this is a reasonable assumption. With a 1-10% duty cycle, different schedules are likely to be in the 99-90% remaining time.

4.6 Multi-hop Cluster Latency

Using ATDMA, intra-cluster packets have a bounded latency of $L_{intra} \leq N_C T_C$ (the same as the single-hop operation). Inter-cluster latency is expressed by:

$$L_{inter} \leq \sum_{i \in \{\text{clusters}\}} L_{i-intra} \tag{2}$$

$L_{i-intra}$ denotes the intra latency of the i-*th* cluster in the path from source (first cluster) to destination (last cluster) (this route is determined by upper level protocols). Let N_H be the number of hops in the path (N_H is always the number of clusters in the path minus one) and N_{C_i} the TDMA frame length of the i-*th* cluster, then:

$$L_{inter} \leq (T_C \sum_i N_{C_i}) \leq (N_H + 1) T_C \max_i \{N_{C_i}\} \tag{3}$$

This relationship holds for any (inter or intra-cluster) data exchange.

For the sake of clarity, let us assume that node A_4 wants to send a packet to node B_2 in the scenario of figure 4. The packet must be delivered through the path $A_4 \rightarrow A_1 \rightarrow B_2$. Intra-cluster A transmission ($A_4 \rightarrow A_1$) maximum latency is given by eq. 1. In the worst case A_4 packet arrives at A_1 just after A_1 slot in the cluster B (as shown in label T3 in the example). Then, a full additional B superframe is required to complete the $A_1 \rightarrow B_2$ transmission (label T4 in the figure). Therefore, in this example, $L_{inter} \leq L_{A-intra} + L_{B-intra} = (4+3)T_C = 7T_C$.

5 Simulation Results

In this section we evaluate VTS through comparative simulations with S-MAC (with and without adaptive listening). S-MAC is chosen as reference because it is a general purpose protocol, it is well documented and previous results can be found in the open literature [2]. All single-hop experiments are evaluated in a 20 node cell for VTS, S-MAC and S-MAC with adaptive listening. All figures show the measured parameter versus the packet Inter Arrival time (IAt, where $IAt = 0$ means that all the packets are generated at the same time). Unicast packet destination is randomly chosen with equal probability among all the neighbors.

Table 1. VTS simulation parameters

Parameter	Value
Radio bandwith	20 Kbps
CTL packet	11 bytes
Listen period	130 ms
Duty cycle	10%
Contention window	31 slots, 1 ms/slot
Consumption in reception state	14.4 mW
Consumption in transmission state	36 mW
Consumption in sleep state	15 μW
Initial N_C counter	20
Inactivity counter (N_I)	5 superframes
Setup cycles (N_S)	20 cycles
T_C	1.3 s

Simulation Configuration. OMNET++ [2] is used as simulation platform. Simulation parameters are selected from reference [2], using the Mica motes [3] as underlying hardware. Table 1 shows main simulation parameters. Aditionally, the following options are set for all the simulations:

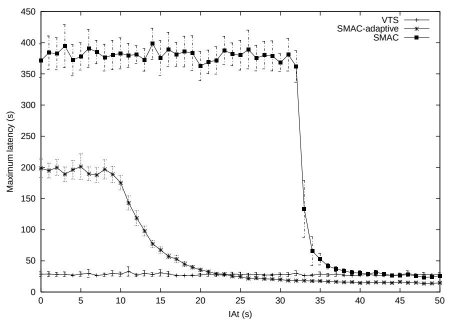
- A simulation finishes when all the nodes have sent 1000 data packets (70% unicast and 30% broadcast). Data packets are 100 bytes long.
- Network packet generation starts after a transient time (100 s) plus a random number of cycles uniformly distributed between 0 and 50. Packet generation is then deterministic: packets arrive after a selected IAt.

Maximum and Average Latency. Figures 5(a) and 5(b) show the maximum and the average latency, respectively. A bounded latency is expected for VTS as discussed in section 4.4. Experiments confirm that in all the cases latency never exceeds the superframe length (as obtained from eq. 3). VTS effectively adapts the frame length to the actual number of nodes present. In comparison, S-MAC maximum packet latency is clearly not bounded, even with adaptive listening.

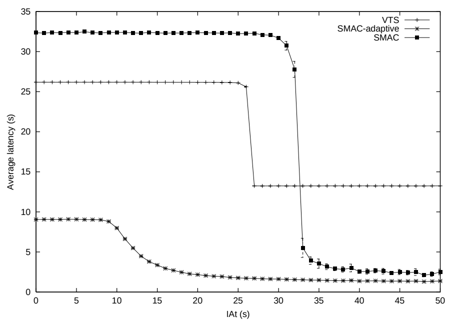
Under high load conditions (low IAt), VTS keeps average latency equal to the superframe length (20 T_C = 26 s). For low and moderate loads (medium and high IAt), the average latency depends on the packet generation time, which is uniformly distributed in the timeslot, yielding a latency reduction of one half. In S-MAC, latency depends on the number of nodes contending for the medium. For

[2] http://www.omnetpp.org
[3] http://www.xbox.com

moderate loads, only a few nodes contend and latency reduces below that of VTS, because VTS nodes must wait for their timeslots to transmit. Adaptive listening is an improvement of S-MAC to reduce latency. Consequently, it outperforms both VTS and S-MAC. Although this is a trade-off between average latency and energy consumption, as it will be shown later.



(a) Maximum latency



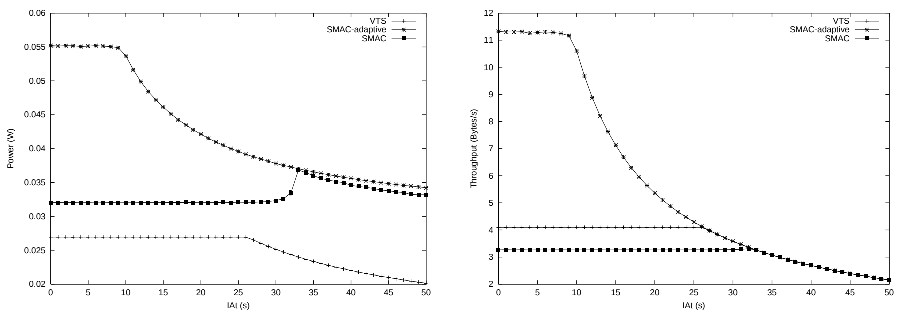
(b) Average latency

Fig. 5. Single-hop configuration (average ± 99% confidence interval)

Power Consumption. Figure 6(a) shows the average network power consumption. S-MAC consumes a 18% more than VTS at high loads (due to the double contention interval per slot and the collisions). S-MAC with adaptive listening consumes two times more power than VTS. This is the cost of reducing latency. In this case, nodes wake up many more times and try to send packets during the scheduled sleep time, consuming the energy of an additional listen interval. As load decreases, the adaptive listening mechanism is not necessary and the behavior is similar to normal S-MAC. In this case, nodes sleep early in VTS, which increases power saving up to 75%. Reduction of VTS power consumption is higher than S-MAC one as load decreases. Under very low loads (high IAt), VTS will significantly increase the network lifetime compared to S-MAC (the peak at IAt=30 is due to a steeped decrease in latency 5(b), since time to deliver all the packets is reduced and power is measured as total energy consumed divided by total time). In conclusion, in VTS there is a trade-off between latency and energy consumption at low loads, while it guarantees latency at high loads.

Throughput. Figure 6(b) shows that at high loads VTS performs slightly better than S-MAC. VTS can handle traffic peaks as properly as a contention protocol. S-MAC with adaptive listening outperforms VTS but at the cost of a higher consumption.

Transient time. An experiment was conducted to evaluate the time needed to reconfigure the timeslot arrangement when nodes appear in the network. In this experiment, a 16 nodes network is set and progressively 4 additional nodes join the cell, causing allocations loops (see Sect. 4.1). The results show that the average transient time until superframe is established again is 30.08 s (slighty higher than a single superframe time, $N_C T_C = 26$ s).



(a) Average power consumption (b) Average throughput per node

Fig. 6. Power consumption and throughput for single hop (average ± 99% confidence interval)

Multi-hop scenario. Scenario of Fig. 4 is set for the multi-hop configuration experiments. That is, two clusters (A and B) with 3 and 2 nodes respectively, plus one common border node (A_1). IAt was set to 0 (always one packet to send), One node of cluster A sends packets to another node of cluster B through the border node. The rest of the nodes send packets to randomly selected intra-cluster nodes. As expected, from 0.15 to 1 s offset there is no overlap between schedules, thus all the experiments exhibit a maximum latency under the upper bound obtained in eq. 3 ($7T_C$ in this case). Average latency depends on the offset between schedules. If there is overlapping, VTS still works and delivers all the packets but latency is not bounded.

6 Conclusions

In this paper we proposed VTS, a protocol for WSNs with bounded latency. VTS dynamically creates a superframe of timeslots and adapts its length to the number of actual nodes in a cell for optimum performance. VTS implements a very simple mechanism to adjust and assign timeslots to the nodes (Sect. 4.1 and 4.2). This protocol may be extended to a multi-hop network if the medium is shared by multiplexation of activity/sleep cycles of clusters (this access scheme is called ATDMA). Its behavior is examined in Sect. 4.5, concluding that, as long as nodes do not overlap, multi-hop operation has a bounded latency. Finally, expressions for the maximum latency of both intra and inter-cluster links have been obtained (Sect. 4.4 and 4.6). VTS further proposes to use a short single control packet to announce any node intentions during its timeslot. Thereby, VTS saves energy by reducing the amount of time a node needs to listen to the channel. Simulations reveal that VTS has an excellent power consumption profile, which is crucial in WSN. Under low loads VTS compromises latency and energy consumption, while it guarantees latency at high loads. Our future work includes the development of a generalized mechanism that ensures proper ATDMA operation (independently of the relative schedule delay among clusters) and an evaluation of scalability. We plan also to implement and test it with real devices.

7 Acknowledgments

This work has been cofunded by the Economy, Industry and Innovation Council, with the SOLIDMOVIL project (2I04SU044), supported by Fundacion Seneca, from the Region of Murcia with the ARENA Project (00546/PI/04), with the ARPaq project (TEC2004-05622-C04-02/TCM) by the Spanish Research Council and the CSI-RHET project (TEC2005-08068-C04-01/TCM).

References

1. Akyildiz,I. F., Kasimoglu, I. H.: Wireless sensor and actor networks: Research challenges. Elseveier Ad Hoc Networks, Vol. 2, issue 4 (1995) 351-367
2. Ye, W., Heidemann, J., Estrin, D.: Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks. ACM/IEEE Transactions on Networking, Vol. 12 (2004) 493–506
3. Stemm, M., Katz, R. H.: Measuring and reducing energy consumption of network interfaces in hand-held devices. IEICE Transactions on Communications, Vol. E80-B (1997) 1125–31
4. Singh, S., Raghavendra, C.: Power efficient MAC protocol for multihop radio networks IEEE Interational Symposium on Personal, Indoor and Mobile Radio Communications, Vol. 1 (1998) 153–157
5. van Dam, T., Langendoen, K.: An adaptive energy–efficient MAC protocol for wireless sensor networks. SenSys 03-ACM Conference on Embedded Networked Sensor Systems (2003) 171–180
6. Zhu, C., Corson, M. S.: A five–phase reservation protocol (FPRP) for mobile ad hoc networks. Kluwer Wireless Networks, Vol. 7, no. 4 (2001) 371–384
7. van Hoesel, L., Nieberg, T., Kip, H., Havinga, P.: Advantages of a TDMA based, energy–efficient, self–organizing MAC protocol for WSNs. IEEE VTC (2004)
8. van Hoesel, L., Havinga, P.: A lightweight medium access protocol (LMAC) for wireless sensor networks. INNS 04-International Workshop on Networked Sensing Systems (2004)
9. Langendoen, K., Halkes, G. Energy–efficient medium access control. In: Zurawski, R. (ed.): Embedded Systems Handbook, CRC Press (2005)
10. Basagni. S.: Distributed clustering for ad hoc networks. Proc. 1999 Int. Symp. on Paralled Architectures, Algorithms and Networks (1999) 310–315