This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Survey on Latency Issues of Asynchronous MAC Protocols in Delay-Sensitive Wireless Sensor Networks

Doudou Messaoud, Djenouri Djamel, and Badache Nadjib

***Abstract*—Energy-efficiency is the main concern in most Wireless Sensor Network (WSN) applications. For this purpose, current WSN MAC (Medium Access Control) protocols use duty-cycling schemes, where they consciously switch a node's radio between active and sleep modes. However, a node needs to be aware of (or at least use some mechanism to meet) its neighbors' sleep/active schedules, since messages cannot be exchanged unless both the transmitter and the receiver are awake. Asynchronous duty-cycling schemes have the advantage over synchronous ones to eliminating the need of clock synchronization, and to be conceptually distributed and more dynamic. However, the communicating nodes are prone to spend more time waiting for the active period of each other, which inevitably influences the one-hop delay, and consequently the cumulative end-to-end delay. This paper reviews current asynchronous WSN MAC protocols. Its main contribution is to study these protocols from the delay efficiency perspective, and to investigate on their latency. The asynchronous protocols are divided into six categories: static wake-up preamble, adaptive wake-up preamble, collaborative schedule setting, collisions resolution, receiver-initiated, and anticipation-based. Several state-of-the-art protocols are described following the proposed taxonomy, with comprehensive discussions and comparisons with respect to their latency.**

***Index Terms*—Wireless Sensor Networks, MAC protocols, Asynchronous Protocols, Quality of service, Delay-sensitive application, Real-time applications.**

I. Introduction

NOWADAYS it has become possible to build tiny, wireless communication enabled, hardware devices, for monitoring and measuring miscellaneous parameters of the environment. This yields wireless sensor networks (WSNs); a special class of wireless networks where nodes are low cost, resource constrained, and generally battery powered devices. In a WSN, nodes are typically deployed in large number throughout a hostile and unattended environment, making battery replacement impractical. Therefore, energy saving would be the main concern for the design of an effective WSN. Current radios consume about as much energy when running idle (switched on, or also called ready-to-receive mode) as when transmitting or receiving data. For example, TI/Chipcon CC2420 consumes about 18.8 mA at idle and receiving mode, and 17.4 mA at transmit mode [1]. Only by putting the radio into (deep) sleep that the energy consumption is reduced considerably. Given that the radio component of a typical sensor node is likely to consume the largest amount of the node's battery. For instance, an AVR Atmel microcontroller runnig at 8 MHz consumes about 4.7 mA [2]. Energy conservation is thus achieved at the MAC layer using duty-cycling of the radio. That is, repeatedly switching it off/on. In active mode, a node can receive and transmit packets. While in the sleep mode, it completely turns off its radio to save energy. In this situation, a node needs to be aware of its neighbors' wakeup time, since packets cannot be exchanged unless both the transmitter and the receiver are awake. This has a direct impact on the forwarding delay of sensed data.

While the early research on WSNs has mainly focused on monitoring applications, such as agriculture [3], and environmental monitoring [4], which are based on low-rate delay tolerant data collection, current WSN applications can support more complex operations, ranging from health care [5], to industrial monitoring, transportation, automation [6], multimedia and visual WSN [7] and [8]. These WSN applications are known as delay-sensitive applications [9], where the timeliness issues is a parameter to be considered. The delay constraints of WSNs will be of increasing importance for some of such applications; notably in early warning systems for disasters, such as fire detection, radiation, ozone or methane emissions, earthquakes, tsunamis, flood, etc. Moreover, real-time constraints may be stringent in applications such as factory automation, health-care systems, ambient assisted living, or intelligent transportation systems.

The provision of delay guarantee in WSN is challenging, due to sensor nodes' limitations in energy supply, computational and communication capabilities, common failures, etc., in addition to the unstable wireless links and the large-scale nature of WSNs. The end-to-end delay is the most critical factor in real-time event-driven applications, where an event needs to be reported to a sink as soon as it is detected, so that the appropriate action can be taken immediately. In periodic monitoring and on-demand traffic applications, real-time communication may not be required, but it might be required that information at the sensor node must reach the sink node within a reasonable time. Thus, most of these applications require an end-to-end delay guarantee for time-sensitive data.

Manuscript received 10 May 2011; revised 7 December 2011 and 11 January 2012.

The authors are with DTISI, CERIST Research Center, 05 Rue des Trois Frère Aissou, Ben-Aknoun, BP 143, Algiers, Algeria (e-mail: {doudou,ddjenouri,badache}@mail.cerist.dz).

Digital Object Identifier 10.1109/SURV.2012.040412.00075

A. Related Work

Delay-aware protocols in WSN, especially routing, have been extensively studied in the literature. State-of-the-art routing protocols can be classified into four categories [10]; i) data-centric, ii) hierarchal protocols, iii) location based protocols, last and foremost, iv) QoS protocols. Protocols in the last class are those that take into account end-to-end delay as a vital routing criterion. Many QoS protocols have been proposed thus far, e.g., Ergen [11]. SPEED [12], Multi-path Multi-speed (MMSPEED) [13], RPAR [14], EA-QoS [15], THVR [16], PATH [17], DARA [18], LOCALMOR [19], and others [20], [21], and [22]. Most of these protocols are surveyed in [23] and [24]. Some routing protocols use cross-layer design by incorporating differentiation services at the MAC layer (prioritized MAC), like RAP [25]. Whereas, other protocols rely on Energy Efficient MAC like B-MAC [26], and try to estimate the single-hop transmission delay so that the minimum-delay routes can be selected. However, estimating the MAC delay at the higher layer may be prone to a high degree of errors. Further, in applications of unpredicted traffic with low duty-cycling, most of the delay is due to receiver sleeping period, i.e., time separating event occurrence and receiver activation. This cannot be tackled by the routing protocol, but it should be dealt by the MAC layer protocol. Several energy-efficient commercial and standardized solutions using cross-layer design and power-efficient MAC protocols have been proposed, such as WOSA [27], KNX [28], IETF 6LowPan [31], IETF ROLL [32], etc. In addition to energy, some other standards consider the communication latency, such as the IEEE 802.15.4 [29], and WirelessHART [30], which is the first open wireless standard for process automation applications.

Lately, numerous surveys on WSN MAC protocols have been published [33], [34], [35], and [36]. The reviewed MAC protocols are evaluated from the energy efficiency perspectives. Kuntz [37] classifies MAC protocols into synchronized protocols, preamble sampling protocols, and hybrid protocols, then he analyzes them from energy efficiency and mobility support point of views. Bachir et al. [38] give a novel taxonomy, in which protocols are classified according to the targeted problems, and they determine the traffic pattern each protocol is suitable for. However, none of the previous surveys traits the timeliness issues of asynchronous MAC protocols. In [39] [40], and [41], thorough analysis of delay guarantee of some energy-aware wireless sensor network MAC protocols is given. Langendoen and Meier [42] investigate the energy and the average latency of some low data rate MAC protocols. Their analysis was limited to Lightweight MAC (LMAC) [43] TDMA-based (Time Division Multiple Access), Scheduled Channel Polling MAC (SCP-MAC) [44] (slotted), WiseMAC [45] (Preamble-based), and Crankshaft [46] (Hybrid) as the best alternative of each class of MAC protocols. However, the authors neither provide a thorough review nor a taxonomy of the literature, but have been limited to comparing these protocols. Suriyachai et al. [47] focus in their survey on two essential network performance metrics; delay and reliability, which are required to supporting mission-critical data delivery. The authors divide mission-critical applications according to these two metrics into, delay-tolerant loss-tolerant, delay-intolerant loss-tolerant, delay-tolerant loss-intolerant, and last but not the least delay-intolerant loss-intolerant applications. They accordingly classify MAC protocols from the delay perspective into protocols that provide node-to-node or end-to-end delay decrease (contention-based), and protocols that ensure node-to-node or end-to-end delay guarantee (contention-free). Following this classification, they gave much more interest to contention-free protocols since they provide deterministic delay/reliability guarantee. However, contention-free protocols, like TDMA, are hard to implement in real motes due to their complexity and sensibility to instable links or frequent topology change.

Depending on how a sender joins its intended receiver, contention-based MAC protocols can be classified into synchronous vs. asynchronous protocols. Synchronous or slotted contention-based MAC protocols specify and coordinate active/sleep periods (duty-cycling schedule). Nodes periodically exchange SYNC packets for synchronization, and they communicate in common active schedule. Synchronization engenders additional charge, and causes more contentions and collisions by grouping communications in small periods. On the other hand, asynchronous low-power protocols have been very common in current WSN deployments. These protocols have no overhead for synchronization, contrary to the synchronous approaches. This explains their wide utilization and implementation in many WSN operating systems, e.g., TinyOS LPL (Low power Listening) [48]. While the synchronous protocols are heavily impacted by clock drift [49], asynchronous protocols experience no such effects. Nonetheless, in asynchronous protocols, communicating nodes are totally decoupled, which may significantly increase the delay for the sender to meet the receiver's active period. This will inevitably influence the one-hop delay, and consequently the cumulative end-to-end delay. In this paper, state-of-the-art energy-efficient low-latency asynchronous MAC protocols are reviewed. More focus will be addressed to timeliness of the considered protocols. The paper proposes a novel classification according to the delay-efficiency-related features of the discussed protocols. The main contribution is to review the state-of-the-art on *asynchronous* energy-aware MAC protocols of WSN, and to accordingly introduce a novel classification on the basis of the mechanisms that affect the delay. On the one hand, our taxonomy is more focusing and is limited to delay-sensitivity protocols (not reliability). On the other hand, it is not limited to a specific class of applications or traffic.

The remainder of the paper is organized as follows. First of all, Section II presents a brief overview of MAC protocols, as well as a taxonomy that is used to categorize the reviewed asynchronous MAC protocols. Next, Section III discusses some design issues of MAC protocols. The introduced taxonomy is described with more details in a subsequent sections. Sections IV and V present static wake-up preamble protocols and adaptive wake-up preamble protocols, respectively. Collaborative schedule setting protocols are presented in Section VI, while collision resolution protocols are presented in Section VII. Section VIII discusses protocols that are receiver-initiated, distinguished from all the other protocols that are sender-initiated. Section IX is devoted to anticipation based protocols. All the reviewed protocols are then summarized in Section X, together with some discussions and open research directions. Finally, Section XI draws the conclusions.

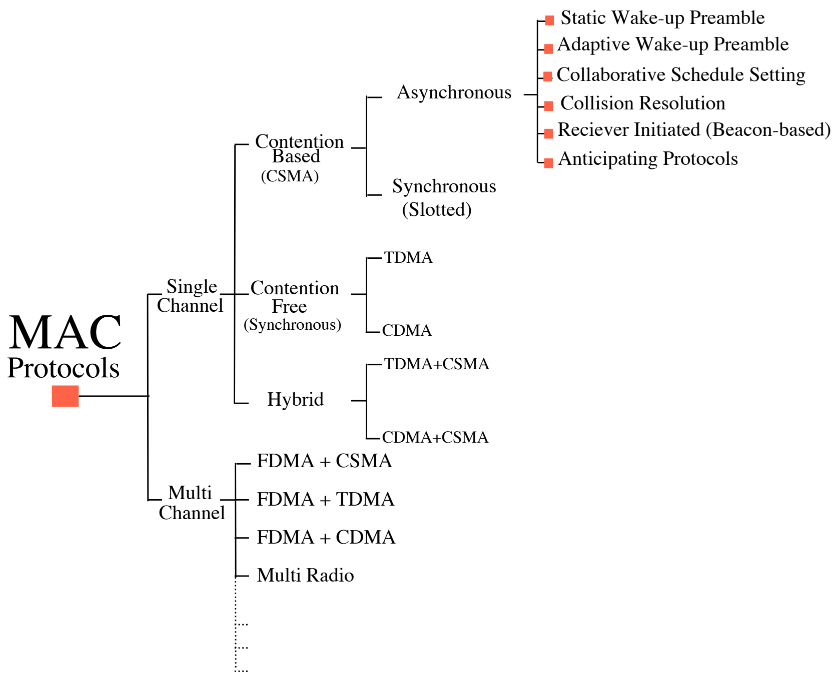

Fig. 1. MAC protocols Taxonomy

II. MAC Protocols: Overview and Taxonomy

A. Existing Standards and Taxonomies

In the literature, MAC protocols are classified into MAC protocols that use a single radio-frequency, vs those using multi-frequency and/or multi-radio [114]. The later aim at improving some network performance metrics, such as latency and throughput. According to the medium access strategy, MAC protocols can be categorized into random access (contention-based), contention-free, and hybrid schemes [39]. Existing contention-based MAC protocols in WSNs can be categorized into synchronous slotted approaches, like S-MAC [50] and DMAC [51], vs asynchronous protocols like B-MAC [26] and X-MAC [52]. The former specify the period of wake-up and sleep for communication, to reduce the unnecessary time in idle listening. Nodes periodically exchange *SYNC* packets for synchronization and communicate in common active/sleep schedule. However, additional tasks would be needed for synchronization, which consumes more energy, computational resources, and bandwidth. On the other hand, asynchronous protocols eliminate synchronization overhead, which improves energy efficiency. However, the sender and the receiver become totally decoupled. This may significantly affect the delay due to the sender waiting for the receiver to wake up.

TDMA-based solutions constitute a class of contention-free MAC protocols. They divide time into slots and use either a centralized or a distributed scheduling scheme. Information related to network topology and traffic load at each node are used to allocate for each one a set of collision-free slots. The schedule determines which node can transmit at a particular time slot, which avoids conflicts and interferences. PEDAMACS [53], RT-MAC [54], Burst [55], GinMAC [56], and WirelessHART [30] are all TDMA-based solutions that consider delay guarantee. Slot allocation and schedule maintaining present a drawback of the use of TDMA in sensor networks. TDMA is sensitive to dynamic networks with frequent topology change caused by the insertion of new nodes, node failure, unstable links, and node mobility. Furthermore, it requires accurate synchronization and must deal with clock drift problems. Although TDMA-like protocols do not suffer from collision, latency is still introduced because of waiting for transmission slots in large scale networks.

FDMA (Frequency Division Multiple Access) and CDMA (Code-Division Multiple Access) are considered as the best alternative to improve throughput and latency in other wireless networks such as cellular networks. Since energy efficiency is the primary goal in WSN, these protocols are unsuitable for this category of networks. FDMA can offer a collision-free medium by dividing bandwidth into channels, where nodes will communicate simultaneously using different channels. Beside this, FDMA eliminates latency introduced by time-based mechanisms. However FDMA requires additional circuitry to dynamically communicate with different radio channels. This increases the cost of a sensor node's hardware. CDMA also offers a collision-free and enables simultaneous-transmission. But its high computational requirement is a major obstacle for use in WSN. EDF [57] and Dual-mode MAC [58] ensure collision-free transmissions using a hybrid FDMA-TDMA scheme. However, this scheme presents the following disadvantages: (1) energy efficiency is not considered, (2) dedicated frequency channels are required for different cells, and (3) a cellular network structure is needed (in a protected mode). Numerous Multi-channel or multi-radio protocols for delay sensitive applications have been proposed thus far, such as T-MALOHA [59], Alert [60], HyMAC [61], RMAC [62], E2RMAC [63]. MCRT as described in [64] and [65] is a multi-channel real-time communication protocol that proposes a flow-based channel allocation strategy to provide delay guarantee data delivery. Multi-channel and multi-radio protocols that employ a wake-up channel or a wake-up radio are out of this paper. For a detailed review of these protocols, one can refer to [38] and [114].

Many hybrid schemes combining TDMA with CSMA have been proposed in the literature, such as Funneling-MAC [66] and Z-MAC [67], but none of these solutions considers the timeliness issue. IEEE 802.11 PSM (Power Save Mode) is an efficient way to save energy; however, it has two drawbacks. First, it is not suitable for multi-hop networks and, second, it introduces latency in the traffic exchange [38]. H-MAC [68] is a recent hybrid protocol that relies on 802.11's PSM mode and slotted aloha to enhance QoS parameters, such as throughput, latency, and channel utilization. But it causes extra overhead during the ATIM (Announcement Traffic Indication Message) period. ATIM is a negotiation widow during which the source node announces the packets to be transmitted, and this requires that all nodes have to be awake and fully synchronized.

Mobility and topology control have also been considered in wireless sensor network MAC protocols [37]. Protocols like MSMAC [69], CFMA [70], and X-Machiavel [71] propose mechanisms that adapt the duty cycle of energy efficient MAC protocols- such as X-MAC and S-MAC- to improve connection setup times and transmission delay, in mobile environments.

Standard groups such as IEEE 802.15, Bluetooth Special Interest Group (SIG), WiMedia Forum, and UWB Forum are working on specifications for the protocols in various scenarios for wireless WPANs and Wireless Sensor networks. The IEEE 802.15.4 standard [29], which is used as a basis for the ZigBee, WirelessHART, and MiWi specifications, has been originally designed for low-rate WPANs. The standard is then adopted by WSNs, interactive toys, smart badges, remote controls, and home automation, operating on license-free ISM bands. IEEE 802.15.4 is intended as a specification for low-cost, low-powered networks. IEEE 802.15.4 can operate in beacon-enabled and in non-beacon mode. In non-beacon mode a CSMA/CA approach is used. It is not collision-free but suitable for many applications. On the other hand, in beacon mode the protocol makes use of a superframe to transmit data. This superframe is composed of three parts; the beacon, a contention access period, and a collision free period (CFP). It is in the CFP part where guaranteed time slots (GTSs) can be allocated in order to transmit data with time critical requirements at the cost of increasing energy consumption. The IEEE 802.15.5 task group specifies mesh networks for WPANs and WSNs as an alternative to IEEE 802.11s (which specifies mesh networks for WLANs). The task group considers both low rate and high rate in their specifications, and it is still active.

B. New Taxonomy

Asynchronous MAC protocols consist a class of CSMA-like protocols that do not restrict which node can access the medium and when. This provides high level of flexibility to handle different nodes densities and traffic loads. Nothing has to be decided prior to deployment, and dynamic changes can be accommodated easily. Moreover, there is no need to synchronize nodes' clocks. Nonetheless, asynchronous random access scheme presents some drawbacks, since the delay is often increased due to several factors, such as rendezvous mechanism, overhearing, over-emitting, and collisions. Because of their energy efficiency, simplicity and flexibility, in addition to their wide utilization in WSN applications, we will focus in this paper on asynchronous MAC approaches. Based on the introduced mechanisms that affect the delay (such as duty-cycling policy), existing asynchronous MAC protocols can be classified into 6 categories namely: static wake-up preamble, adaptive wake-up preamble, collaborative schedule setting protocols, collision resolution protocols, receiver-initiated (which is beacon-based), and last but not least anticipation-based protocols (described with more details in Sections IV to IX). Fig. 1 shows a graphical representation of MAC protocols solutions and the state-of-the-art classification, as well as the novel taxonomy of asynchronous MAC inside this classification. For a complete MAC protocols overview, one can refer to the online survey of energy-efficient MAC protocols available at [72].

III. MAC Protocols Design Issues: Constraints and Requirements

Due to the limited capabilities of WSN, MAC protocols are influenced by a number of constraints, while they should meet specific performance requirements. MAC protocols in WSN are generally application dependent. An effective MAC protocol should consider a set of performance factors and makes tradeoff among them. The most important performance factors that are required for wireless sensor protocols are described in [10], [33], [34], [50]. At the MAC layer, there are many factors affecting both energy and delay. These factors are generally related to the radio mode, the medium access technique, and the time service. Factors such as overhearing, over-emitting, collisions, and control packets overhead affect energy saving as well as forwarding delay. On the other hand, there are factors that affect inversely energy and delay. Factors such as high duty-cycling and idle listening, contribute in energy wasting, while low duty-cycling and sleep mode of a radio cause end-to-end delay degradation. In the following, we focus on those related to the delay.

A. Factors Affecting Delay

Duty-cycling: Numerous mechanisms are employed to optimize energy consumption, such as optimizing the sensing coverage and the network topology, controlling the transmission power, and notably duty-cycling. Given that the most amount of energy is supposed to be devoted to communication, duty-cycling the radio is the most relevant technique to achieve power efficiency. Duty-cycling is a MAC layer mechanism in which, nodes that are not involved in communications turn off their radios and enter into sleep mode. Nodes then periodically wake-up to send data or to check channel activity for possible packet reception. However, low duty cycle usually causes performance degradation in latency. There is often a tradeoff between energy and latency when talking about the degree of duty cycling.

Capacity: Network traffic load beyond the node queuing capacity can raise packet rejection, which contributes in increasing the end-to-end delay.

Overhearing: This happens when a node hears a preamble or a data packet for which it is not the intended destination.

Over-emitting: Extra delay is engendered if a node sends data and the destination node is not ready to receive it. This does not happen with synchronous protocols, but only with asynchronous ones.

Collision: Collision can occur when two nodes transmit packets at the same time and interfere with each other's transmission. Consequently, the delay increases due to the Backoff procedure and retransmissions.

Traffic fluctuations: When a protocol is optimized to a specific traffic, traffic fluctuations can lead to delay degradation. Therefore, the protocol should be traffic adaptive if designed for traffic-diverse applications.

Control packets overhead: Nodes use control packets (SYNC, RTS, CTS, ACK) before sending the data, which help to regulate access to the transmission channel. Sending and receiving control packets incur some delay.

B. Traffic patterns

It is important when studying the behavior of MAC protocols to illustrate the kind of traffic they have to handle. Since MAC protocols in WSNs are generally application-dependant, three traffic patterns can be distinguished:

Local traffic: This includes in-network processing and data aggregation. It involves local messages being exchanged between nearby nodes and their cluster head. Data exchange can be latency-sensitive.

Sensor-to-sink traffic: Periodically or after event occurrence, the sensor nodes need to report data packets to the sink. However, when data packets are delay-sensitive, the reporting should be performed within the designated time so that all data remain valid.

Sink-to-Sensor/Actuator traffic: This pattern includes configuration packets from an upper layer such as (network and application layers) originated from the sink to sensor nodes, as well as actuation tasks from the sink to actuators. In fact, the sink must take a decision after an event occurrence and send it back to the appropriate actuator in real-time so that an action can be immediately triggered. Here, real-time communication may be essential.

C. The Forwarding Delay

The end-to-end delay is defined as the delay from the time when a source node has a data packet ready to send (example: when detecting an event and generating the event reporting packet or packets) to the time the first packet is received at the destination node. For applications that use a single packet to carry the event information, the above definition captures the actual delay for reporting the event information. With comparison to applications that use multiple packets, if the nodes that relayed the first packet stay awake for forwarding all the packets, the delay to relay subsequent packets will be much smaller than that experienced by the first packet. Hence, the actual event-reporting delay can still be approximated by the delay experienced by the first packet. But this is not the case when duty-cycling at the MAC layer is completely decoupled for the higher-layers data traffic.

The inherent properties of WSN constitute an unfavorable environment for delay guarantees. Delay guarantee is achieved through providing deadlines to each individual message at the network and the application level [73] and [74]. Implicit assumptions on the underlying models related to static and regular topologies, symmetry of the radio propagation patterns, absence of environmental interferences, and bidirectional communication patterns when designing low-latency protocols are generally introduced. Authors in [75] present a novel concept of delay guarantee, in which the desired end-to-end delay distribution is expressed by the application as the acceptable interval for sequences of messages to be delivered with the defined level of confidence (probability), rather than the classical strict end-to-end deadlines for individual messages. This way, real-time guarantee can be expressed in terms of QoS, and two main types of real-time (RT QoS) guarantee can be distinguished: hard real-time (HRT) and soft real-time (SRT) [76]. In HRT, deterministic end-to-end delay bound should be assured. The arrival of a message after its deadline is considered as failure of the system. While in SRT system, a probabilistic guarantee is required and some delay is tolerable.

In general the delay is influenced by a set of parameters such as the distance and hop-count toward the sink, the node density, data rate, node's resources (energy), MAC, and routing communication protocols. All these parameters can provide high and unpredictable end-to-end delay. From the layer stack point of view, the MAC layer determines the one-hop delay since it controls the medium access. Whereas, the network layer controls the multi-hop delay since it is responsible for route selection. As routing and MAC are both affecting the end-to-end delay, the cross-layer design would be a promising solution.

At the MAC layer, the one-hop delay can be expressed as:

$$\begin{aligned} OneHop_{delay} &= Processing_{delay} + Queuing_{delay} \\ &+ Channel\_Access_{delay} + Transmission_{delay} \\ &+ Propagation_{delay} + Reception_{delay} \end{aligned} \tag{1}$$

The end-to-end delay, which is the accumulation of the previous metric, can be expressed at the network layer as:

$$EndToEnd_{delay} = \sum_{i=1}^{k} OneHop_{delay}(i) \tag{2}$$

where i $\in$ {1,.., k} is the individual link of the selected route. Given that transmission, reception and propagation delays are hardware dependent, and more latency can be engendered when queuing packets and accessing the channel, current researches focus on improving queuing strategies and channel access techniques to ensure QoS MAC requirements in terms of energy and delay efficiency. This paper considers an arbitrary network topology where packets are forwarded by node TX_i and received by RX_i at a given hop h_i. The process is repeated until the packet reaches its destination sensor node or the sink. This uses the many-to-one communication pattern that is commonly adopted in several WSN deployments. This forwarding scenario is adopted to illustrate operations of the

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

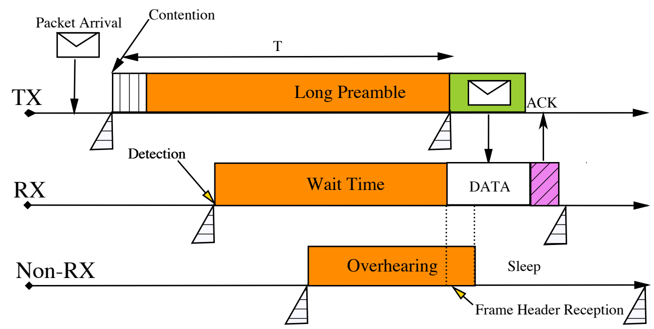

Fig. 2. B-MAC's Long Bit-stream Wakeup Preamble

reviewed protocols. A hop-by-hop analysis of the reviewed MAC protocols from delay performance point of view is provided, to investigate whether the protocol achieves delay guarantee or delay decrease. The remaining of the paper presents each class of the proposed taxonomy, and it appropriately discusses state-of-the-art protocols. Because of the space, this survey cannot cover all MAC protocols proposed in the literature. But typical protocols are included and discussed in each section.

IV. Static Wake-Up Preamble Protocols

In static wake-up preamble protocols, the employed preamble is fixed in advance and does not change according to the neighboring schedule or the network traffic load. Fixing the length of wake-up preamble makes the protocol rather simple. But this makes it suffers from rigidity to support delay-sensitive applications. The typical protocol that uses static wake-up preamble is B-MAC proposed by Polastre et al. [26].

B-MAC [26] is considered as a canonical energy efficient MAC protocol, and it is used herein as a baseline reference for protocol comparison. B-MAC is the first LPL (Low Power Listening) protocol in which a node utilizes pure physical-layer information from the radio to quickly detect ongoing transmissions, and to wake up during one of its periodic receive checks. Nodes simply reverse the role of clear channel assessments, which allows them to rapidly return to deep-sleep if no channel activity is detected. The use of layer 1 requires B-MAC's nodes to remain awake throughout the duration of long-preamble transmissions before obtaining the useful data packet (see Fig. 2), since there is no mean to learn the link-layer address of the target. EA-ALPL [77] and SEESAW [78] optimize B-MAC by setting node's listening mode according to its current and past forwarding loads.

Since B-MAC uses long preamble, it causes extra latency at each hop. Furthermore, the use of periodic layer 1 receive checks obligates B-MAC transmitters to send packets using long bit-stream preambles equivalent to one wake-up interval, without target ID. These preambles wake-up every node within vicinity of the transmitter, whether or not it is the target. This results in the well-known overhearing problem. The expected B-MAC's one-hop delay is given by Eq. (3):

$$OneHopDelay_{B-MAC} = t_{Contention} + t_{LongPreamble} + t_{Packet} + t_{Ack} \tag{3}$$

TABLE I
B-MAC's Delay Symbols

Symbol	Meaning
$t_{Contention}$	Contention period including(t_{cs}: Carrier Sensing and Backoff Period)
t_{Packet}	$(Packet\ length)/(Data\ Rate)$
t_{Ack}	$(Acknowledgement\ packet\ length)/(Data\ Rate)$
$t_{LongPreamble}$	$= Channel\ sampling\ period$

If the wake-up interval of B-MAC is T, then the average wake-up preamble time is: $t_{LongPreamble} = T/2$.

Table I explains different Symbols composing the B-MAC's delay. Note that in the rest of the paper, two successive frame transmissions are always separated by short time called SIFT (Short Inter Frame Space), and must be taken into account when calculating transmission delay.

Protocols like X-MAC [52], DPS-MAC [79], Patterned Preamble [80], AREA-MAC [81], CSMA-MPS [82], TICER [83] and MH-MAC [84] use the same principle to cope with long preamble introduced by B-MAC. They probe the receiver using repeated short preambles to reduce preamble overhead saving energy and to give other nodes the chance to forward their packets quickly. X-MAC improves B-MAC by using only link-layer information in its approach for channel polling (periodic checking for channel activity), and replacing long preambles by short ones. The sender will send out the strobe short preamble (equivalent to RTS) with the embedded target address, so that the non-target receiver can quickly go back to sleep. This scheme addresses the overhearing problem and saves the energy consumed by the non-target receivers. By using the strobe preamble, the target receiver can send back an early ACK (equivalent to CTS) as soon as it wakes up. This approach reduces the time and energy wasting. X-MAC's layer 2 receive check strategy leaves the radio in receive mode long enough (twice the amount of time required for a transmitter to send a single wake-up packet) to detect the presence of small wake-up packets. X-MAC's delay is improved over B-MAC's delay since the receiver can inform the sender about its wake-up. But the sender, and related to the number of strobes preambles iterations required for the destination to wake-up, still monopolizes the channel as long as the receiver sleeps (on Fig. 3, TX_2 cannot send its preamble unless TX_1's transmission has completely finished), and consequently affects the delay of any other concurrent transmission in the neighborhood.

The expected one-hop latency of X-MAC is given by Eq. (4):

$$OneHopDelay_{X-MAC} = t_{Contention} + (t_{StrobePreamble} + t_{Listen\_Ack}) \times ENIR + t_{Contention} + t_{Packet} \tag{4}$$

$ENIR$: expected number of (Send strobe preamble/ Listen Early_Ack) iterations required. It is proportional to the receiver duty-cycle period.

t_{Listen_Ack}: the time required to receive the $Early\_ACK$ sent by the receiver at its wakeup. The second contention period is used to give other concurrent senders a chance to send data destined to this target node.

STEM [85] has two versions, STEM-T (Tone) which operates under B-MAC mechanism and STEM-B (Beacon), which

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

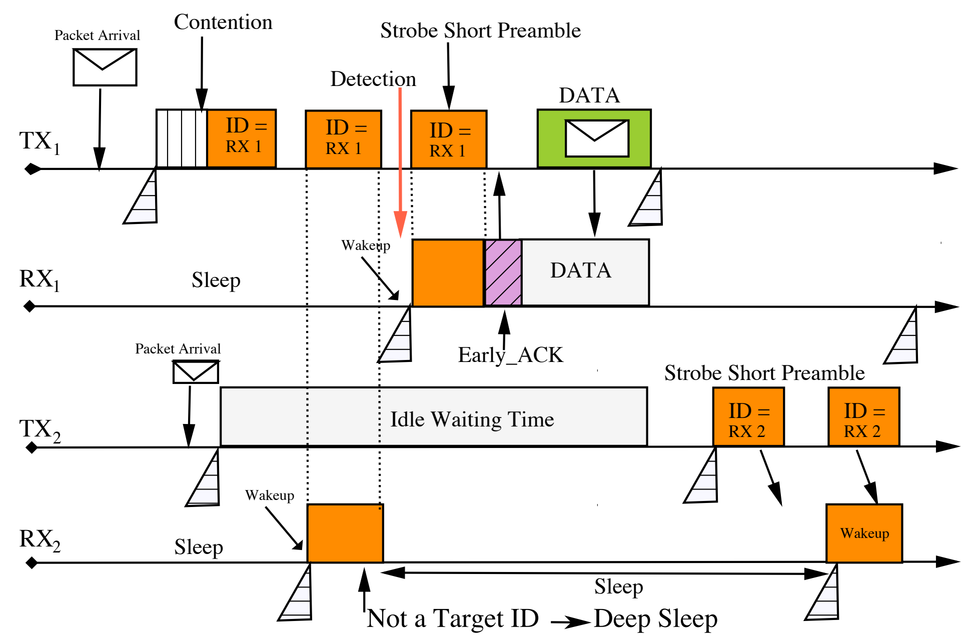

Fig. 3. X-MAC's strobe short preambles.

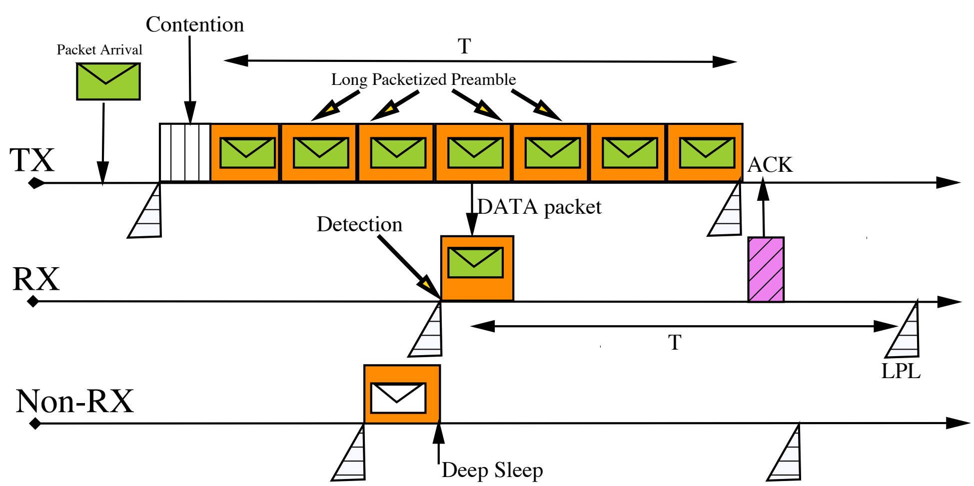

Fig. 4. BoX-MAC-1's long packetized wake-up preamble.

operates under X-MAC mechanism. The key difference is that preamble and data communications operate on two different channels.

BoX-MAC [86] is yet an asynchronous scheme designed with two variants; BoX-MAC-1 and BoX-MAC-2 which are based on B-MAC and X-MAC respectively. SpeckMAC-D [87] and MX-MAC that is introduced recently by Merlin and Heinzelman [88] propose exactly the same scheme as BoX-MAC.

BoX-MAC-1 continuously transmits a preamble containing data packet for a period of T as shown on Fig. 4 instead of long bit-stream preamble. This continuously packetized wake-up preamble allows BoX-MAC-1 nodes to save energy and reduce delay compared to B-MAC by only staying awake to receive significant packets destined to them.

The expected one-hop delay of $_{BoX-MAC-1}$ is given by Eq. (5):

$$\begin{aligned} OneHopDelay_{BoX-MAC-1} &= t_{Contention} \\ &+ t_{Long\_Packetized\_Preamble} + t_{Ack} \end{aligned} \tag{5}$$

$t_{Long\_Packetized\_Preamble}$: channel sampling period.

BoX-MAC-2 uses a low-power link-layer mechanism, but incorporates a very small amount of layer 1 information, rather than waking up long enough to hear a complete packet, as X-MAC does. It examines whether there is energy on the channel by employing physical-layer receive check, which diminishes receive check time by a factor of 4 ([86]). As shown on Fig. 5, by incorporating data into strobe preamble assuming that the data packet is small enough to be used as preamble, this

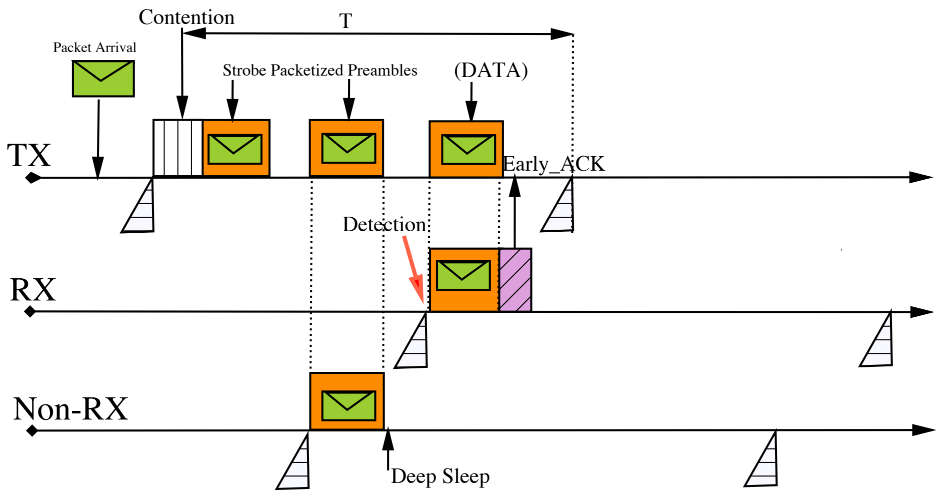

Fig. 5. BoX-MAC-2's strobe packetized wake-up preamble.

permits BoX-MAC to reduce delay compared to B-MAC and X-MAC, since additional time for data packet transmission is eliminated. The expected one-hop delay of BoX-MAC-2 is given by Eq. (6):

$$\begin{aligned} OneHopDelay_{BoX-MAC-2} &= t_{Contention} \\ +(t_{Packetized\_Preamble} + t_{Listen\_Ack}) &\times ENIR + t_{Ack} \end{aligned} \quad (6)$$

$t_{PacketizedPreamble}$: data packet transmission delay.
$ENIR$: expected number of required iterations.

This approach of using data packet as preamble may be suitable for WSN applications that generate very small data packets. But it is definitely unsuitable for the ones with large traffic such as multimedia applications.

RA-MAC [89] uses a succession of small preambles (*RTS* packets) similar to the X-MAC's short preamble in order to reduce the unnecessary transmission delay caused by B-MAC's long preamble. When a sender wants to send, it waits for a random *Backoff* time and then senses the channel. If no activity is sensed, the node transmits an RTS packet and waits for a *CTS* packet during $T_{waitCTS}$. If *CTS* is received, the sender sends *DATA* packet; otherwise, it repeats the same process until D_{SLEEP}.

As an improvement over X-MAC, RA-MAC proposes *RTS* aggregation scheme to integrate *RTS* packets from multiple sender nodes to a single *RTS* packet. From Fig. 6 when a sender *TX(i)* has a packet to send, it monitors the channel, waiting for an *RTS* with the same target address. If no activity on the channel is sensed, the node sends the *MainRTS* packet and waits for the *CTS* packet or the *SubRTS* packet. Once a sender *TX(j)* receives the MainRTS packet from *TX(i)*, it checks if the target address is the same. If it is the case then, it unicasts the *SubRTS* packet to *TX(i)* for packet aggregation purpose. When *TX(i)* receives the *SubRTS* from *TX(j)*, it updates the *MainRTS* packet and rebroadcasts it. When the receiver wakes-up and receives the *RTS* packet, it establishes the data transmission schedule for multiple *RTS* nodes and sends the aggregated *CTS* packet with the embedded information about data transmission schedule. Consequently, each sender starts sending *DATA* at its scheduled time.

RA-MAC does not implement data *ACK*. However, collision can occur between senders with different targets or between *CTS* and *SubRTS*, which is difficult to be detected. This dramatically rises the delay before the packet can reach the destination. Furthermore, if a node has a *DATA* burst; it can't send all its packets in the same schedule. RA-MAC's one-hop delay is the X-MAC's one-hop delay increased by the forwarding delay of neighbors' *subRTS* packets. So the cumulated delay of the packets sent by neighbors that beneficiate from *RTS* aggregation mechanism can affect significantly the end-to-end delay of the first aggregator. For example- refer to Fig. 6- packet transmitted from *TX(j)* can result in increasing the first hop delay of *TX(i)*, as it is sent before forwarding *TX(i)*'s packet by the next hop node. Although static preamble MAC protocols are quite simple for implementation, they are sensible to frequent change in the network traffic load, since they do not adapt their preamble to the dynamic wake-up rate of the neighbors. This may impact the transmission delay of a node. Adaptive wake-up preamble protocols are proposed to cope with the delay engendered by static preamble protocols.

V. Adaptive Wake-Up Preamble Protocols

In adaptive wake-up preamble mechanism, the employed preamble can change dynamically according to the neighbors' schedule and to the network traffic load. By doing this channel overhead caused by long preamble or even repeated small preambles is eliminated. Instead, a node wakes-up just in time before its receiver' wake-up to send a small preamble improving thus the overall latency in the vicinity of a node. This technique was first employed by El-Hoiydi et al. in WiseMAC [45].

WiseMAC [45] is based on short asynchronous duty cycle preamble sampling scheme. The idea is to learn the direct neighbors' sampling period in order to use a wake-up preamble of minimized size. When the receiver's wakeup pattern is still unknown, the duration of the preamble is equal to the full basic cycle duration, say *T*. The own schedule offset is then added to the frame and transmitted to the receiver. Each receiver adds its own schedule to the successful received

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

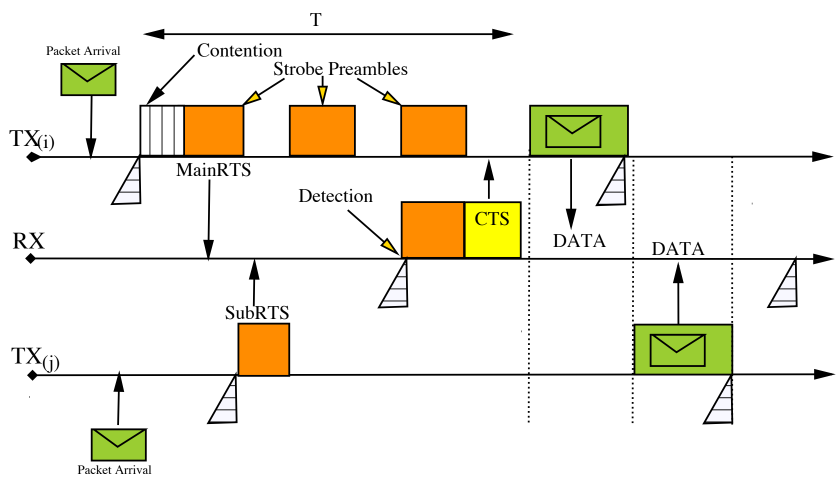

Fig. 6. RA-MAC's RTS Aggregation Scheme.

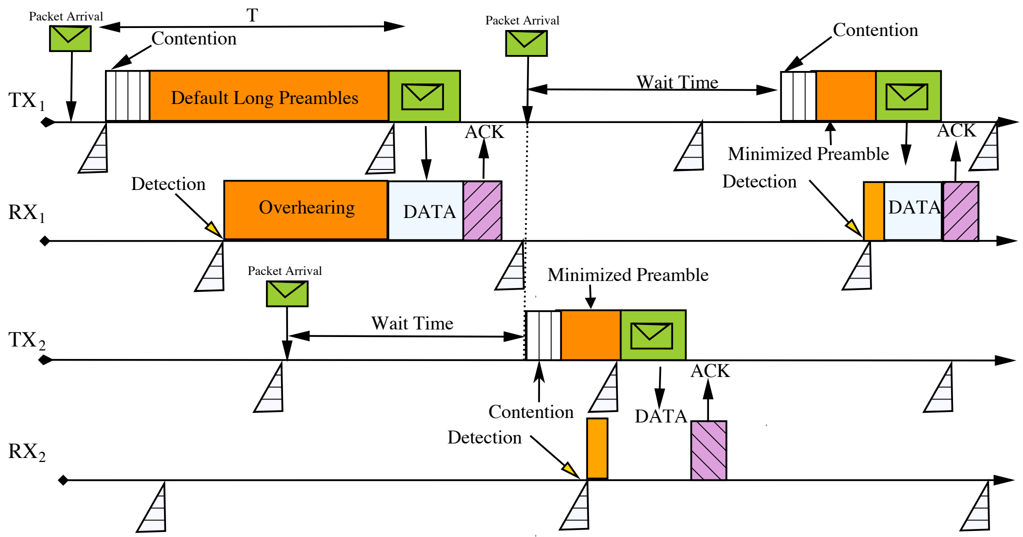

Fig. 7. WiseMAC's Minimized Preamble Size.

TABLE II
WISEMAC'S OFFSET SCHEDULE TABLE

Node ID	Remaining time until the next sampling time
1	$T_{WakeUp}(1)$
:	:
:	:
i	$T_{WakeUp}(i)$

frame's acknowledgement. Received schedule offsets of all neighboring nodes are subsequently kept in a table (as shown in Table II) and are dynamically updated whenever frames and schedules are exchanged or possibly overheard. Based on the schedule offset table, a node can determine the wake-up time of all its neighbors. So it wakes-up at the exact moment to send data, which leaves the channel free for as long as possible, improving thus transmission delay of its neighbors. Following Fig. 7, using minimized preamble size, TX_1 allows TX_2 to send its packet quickly without additional waiting time. The duration of the wake-up preamble must cover the potential clock drift between the clock at the source and at the destination. This drift is proportional to the time since the last schedule exchange. If nodes are equipped with quartz based clock with a drift tolerance of $\pm\theta$, and if L denotes the interval until next communication time, then the duration of the wake-up preamble is given by Eq. 7:

$$T_{Preamble} = min(4\theta L, T) \tag{7}$$

If TX has received RX's schedule timing at time t_0, and that it wants to send the latter a packet at the sampling time L, then its clock will have an advance of θL at time L. Because both clocks of both RX and TX might be either advanced or delayed, TX must target a time $2\theta L$ later or $2\theta L$ early to L respectively. So the duration of the wake-up preamble must be $4\theta L$. If L is very large, then $4\theta L$ may be larger than the sampling period **T**. Thus $T_{Preamble} = min(4\theta L, T)$. The transmission will be started at time $L - \frac{T_{Preamble}}{2}$, to center the wake-up preamble on the expected scheduled sampling time.

WiseMAC uses more-bit in the header of data packets. When this bit is set to 1, it indicates that more data packets are waiting to be sent. The receiving node then continues to listen

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

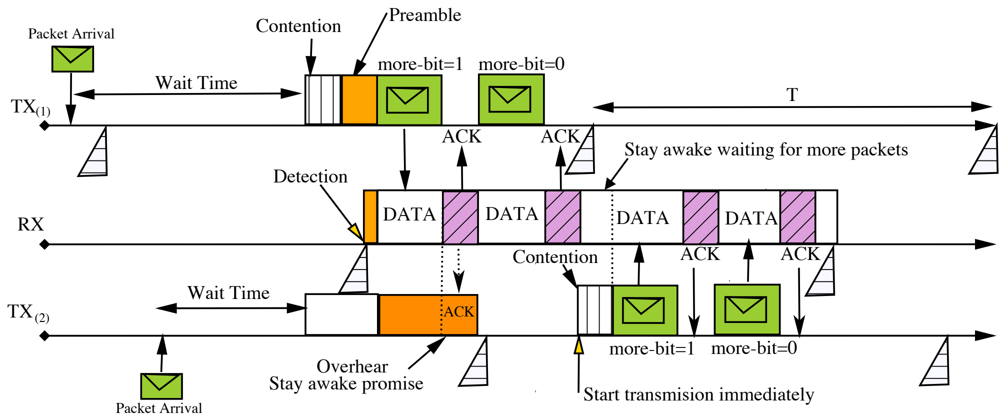

Fig. 8. Extended-WiseMAC with more-Bit and Stay-Awake promise.

after having sent *ACK*. The sender will transmit the following packet right after having received the *ACK*.

UBMAC [90] uses the same scheme as WiseMAC, which it applies to B-MAC. It embodies a hybrid synchronous and asynchronous mechanism, and relies on a synchronization protocol called RATS (Rate Adaptive Time Synchronization). A weak synchronization is maintained by transmitting dedicated synchronization beacons. Synchronization precision is thus traded for reducing communication overhead of RATS. Both WiseMAC and UBMAC use estimate on the neighboring node's wake-up, and accordingly set the preamble size to mitigate possible errors. However, this estimation is refreshed implicitly whenever data exchange takes place in WiseMAC, contrary to UBMAC that uses a synchronization protocol (RATS).

Extended WiseMAC more bit scheme [91] is an optimization over WiseMAC. It allows a common destination node to automatically stay awake for a longer time than just the wake-up period when more traffic has to be handled, which improves the delay. If both $TX_{(1)}$ and $TX_{(2)}$ aim to reach *RX* in the same wake-up interval, the contention mechanism of WiseMAC will decide who is the first to transmit. From Fig. 8, $TX_{(1)}$ wins the contention and sends its first two frames with the more bit set. The destination node acknowledges the more bit in the *ACK* packet and stays awake for at least a basic sampling interval **T**. As $TX_{(2)}$ has lost the contention, it will wait and overhear the transmission from $TX_{(1)}$ to *Rx*. By hearing the Stay-Awake promise in the *ACK*, $TX_{(2)}$ knows that it can start sending its own data frames right after $TX_{(1)}$ has finished its transmissions.

The expected one-hop delay of WiseMAC for the first packet is given by Eq. (8):

$$OneHopDelay_{WiseMAC} = t_{Waiting} + t_{Contention} + t_{WiseMAC\_Preamble} + t_{Packet} + t_{Ack} \tag{8}$$

$$\begin{aligned} OneHopDelay_{Burst\_Packets} &= OneHopDelay_{1^{st}_{packet}} \\ &+ \textstyle\sum_{i=2}^{k=last} OneHopDelay_{i^{th}_{packet}} = t_{Waiting} + t_{Contention} \\ &+ t_{WiseMAC\_Preamble} + t_{Packet} + t_{Ack} \\ &+ \textstyle\sum_{i=2}^{k=last} (t_{Contention} + t_{Packet} + t_{Ack})_{i^{th}_{packet}} \end{aligned} \tag{9}$$

$t_{Waiting}$: the calculated time required for the destination node to wake-up.

In WiseMAC, the sender node during $t_{Waiting}$ can send or receive other packet to/from other neighbors. This can significantly improve the delivery delay of nodes with burst data (nodes that have event containing multiple packets). Furthermore, the delay can be more reduced in Extended WiseMAC using Stay Awake promise, which gives a chance for neighboring nodes to quickly forward their packets. However, the drawback of WiseMAC is that the length of the preamble increases inversely with the communication rate between two neighboring nodes, which makes the one-hop delay converge to that of B-MAC in low-traffic applications.

MaxMAC [92] aims at achieving maximal adaptivity by adjusting the duty-cycle of a node according to the traffic load. It attempts to reach the throughput and latency of energy unconstrained CSMA by allocating extra wake-up periods when the rate of incoming packets reaches the predefined threshold. MaxMAC uses the preamble sampling technique of WiseMAC, enriched with target ID information to ovoid overhearing. It defines 3 thresholds $(T_1 = 2, T_2 = 6, T_{CSMA})$ according to which a node can be in one of the four duty cycle transition states depicted on Fig. 9. In MaxMAC, the request for more wake-ups is receiver-based notification and triggered after a defined threshold, thus first packets may suffer from delay degradation, which is not suitable for event driven WSN.

AS-MAC [93] avoids overhearing and offers adaptive sampling period by employing successive preloads, each containing the address of the target, and the remaining time until data transmission. Preload messages prevent non-target nodes from consuming energy caused by overhearing. At the sender side, the carrier sensing is performed to check channel activity. If the channel is clear, the sender transmits the preload frame during preload period (equal to default sampling period *T*) to cover the forwarding node's wake-up, followed by the data frame. If the sender has burst data packets, it requests the receiver to reduce sampling period using *RSP* field in a Data frame. At the receiver, if a node does not have any data to send, it wakes-up every sampling period to check channel activity. If it hears the preload message, it identifies destination address. If the packet is targeted to itself, it checks the duration value.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

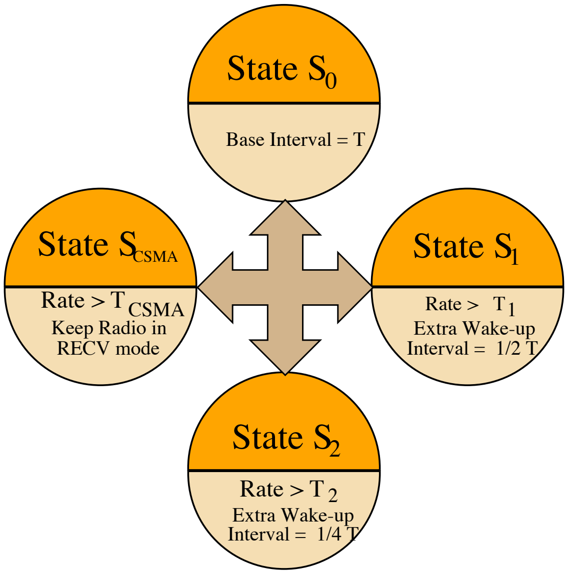

Fig. 9. MaxMAC's Duty-cycle transition states.

The receiver RX on Fig. 10 can immediately go to sleep mode until the end of this duration to receiver data. The node sends back *ACK* and if *RSP* field is set, the receiver adds a new entry in the sampling table to save the sender address and the sampling period. Then, it readjusts its sampling period only if the newly requested sampling period is shorter than the recently used one. That is, a receiver always uses the smallest period in the sampling table.

AS-MAC applies flexible preloads and sampling period to lower latency, by using *RSP* field. The source node requests all forwarding nodes in a path to reduce both channel sampling (Sleep/Active rate) and preload period to the new value indicated in *RSP* field. This is only when the node has more packets to send. The first data packet may suffer from large delay. However, the following data packets can be immediately forwarded, but it is not specified how the new value is calculated. Also, repeating the preloading and sampling when a node has more data to send is time and energy consuming. The expected delay of AS-MAC is given by Eq. (10):

$$\begin{aligned} OneHopDelay_{AS-MAC} &= t_{Contention} \\ &+ t_{Preload\_Frame} + t_{Packet} + t_{Ack} \end{aligned} \tag{10}$$

In a case of single traffic in the network we can calculate the end_to_end delay as in Eq. (11):

$$\begin{aligned} EndToEndDelay_{AS-MAC} &= t_{Contention} + t_{Default\_Preload\_Frame} \\ &+ t_{Packet} + t_{Ack} + \textstyle\sum_{i=2}^{k=Last\_Hop} (t_{Contention} \\ &+ t_{New\_Preload\_Frame} + t_{Packet} + t_{Ack})_{i^{th}_{packet}} \end{aligned} \tag{11}$$

$t_{Default\_Preload\_Frame}$: default channel sampling period.

In ADCA [94], each node sets its own wake-up schedule independently. It is composed of repeated cycle periods of fixed size; each consists of a *contention* period, a *SYNC* period (not for synchronization but to announce the next scheduled time), an *extended*, period and a *sleep* period as shown in Fig. 11. When a node starts-up, it first decides its own sleep/wakeup schedule. Then it broadcasts its schedule and collects all neighbors' schedules. The extended period gives other transmitters which have lost contention in a first contention period (TX_2 on Fig. 11) a chance to forward their packets without waiting for the next cycle improving thus the transmission delay. However in ADCA, nodes must wake-up in each neighbor receiving period to learn its new schedule offset wasting thus energy. Because schedules are set independently, two neighbors' schedules can overlap and hence cause collision. Consequently, the delay performance is significantly affected.

The expected delay of ADCA is expressed by Eq. (12):

$$\begin{aligned} &OneHopDelay_{ADCA} = t_{Waiting} + ((t_{Contention} + t_{RTS}) \times ENIR \\ &+ t_{CTS} + t_{Packet} + t_{Ack}) \times P_{Contention\_success} + (t_{ADCA\_Contention\_Period} \\ &+ t_{SYNC} + (t_{Contention} + t_{RTS}) \times ENIR + t_{CTS} + t_{Packet} + t_{Ack}) \times (1 - P_{Contention\_success}) \end{aligned} \tag{12}$$

$ENIR$: the expected number of required iterations.
$t_{Waiting}$: waiting time until the receiver wakes-up.
$t_{ADCA\_Contention\_Period}$: is the first period of ADCA's three period (Contention, SYNC, and Extended).

In [38], the authors have introduced other protocols with minimized preamble size that are similar to WiseMAC, such as SP (Sensornet Protocol) [95], SyncWUF [96], and RATE EST [97]. These protocols provide the same one-hop delay as WiseMAC. Although adaptive wake-up preamble protocols eliminate channel overhead incurred by static preamble protocols like B-MAC, they are still exposed to waiting time -even in sleep state- for their intended receiver to wake-up. This adds extra latency at each hop because the wake-up schedule of each node is set independently. Collaborative schedule setting protocols are proposed to cope with this waiting time.

VI. Collaborative Schedule Setting Protocols

In this class of protocols, a node chooses its own schedule after receiving all its neighbors' schedule at start-up stage, for the purpose of minimizing the end-to-end packet delivery delay. By doing this a node can set its wake-up time to coincide with that of its forwarders minimizing thus The waiting time of adaptive wake-up preamble protocols. This strategy was employed by the following protocols:

LL-MAC [98] uses an asynchronous scheme where the node broadcasts *ASYNC* message, in which it records its own schedule at starting phase. When a neighbor receives this schedule it stores it, and generates its own schedule by modifying the schedule to a stagger one. When a node receives more than one schedule, it will choose the first schedule it receives as the reference. LL-MAC uses the stagger active schedule to reduce sleep delay in a multi-hop transmission. If d is one-hop transmission delay then, the node *B's* sleep time will be shifted by a period of d from node A, and such will be node C's sleep time from node B, etc. (Fig. 12). Note that this rule only applies for intermediate nodes beyond the first hop, i.e. nodes on the route except the source and its direct neighbors. A node obtains its neighbor's sleep time from the *ASYNC* message. It adds the hop information

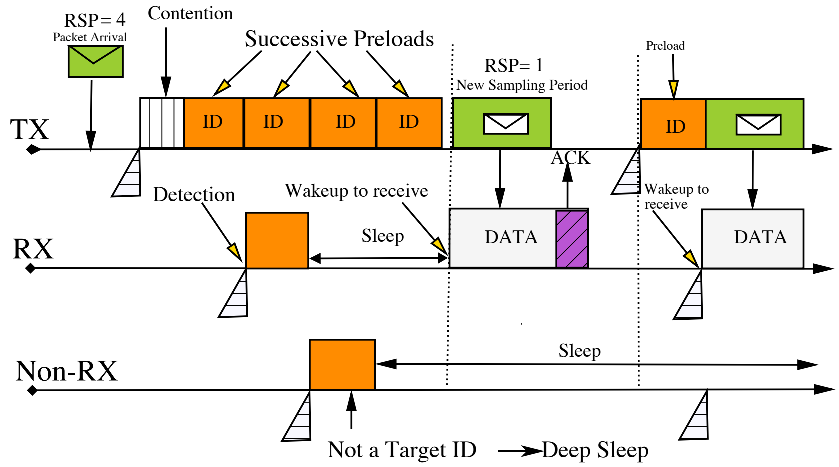

Fig. 10. AS-MAC successive preloads and Adaptive Sampling Period.

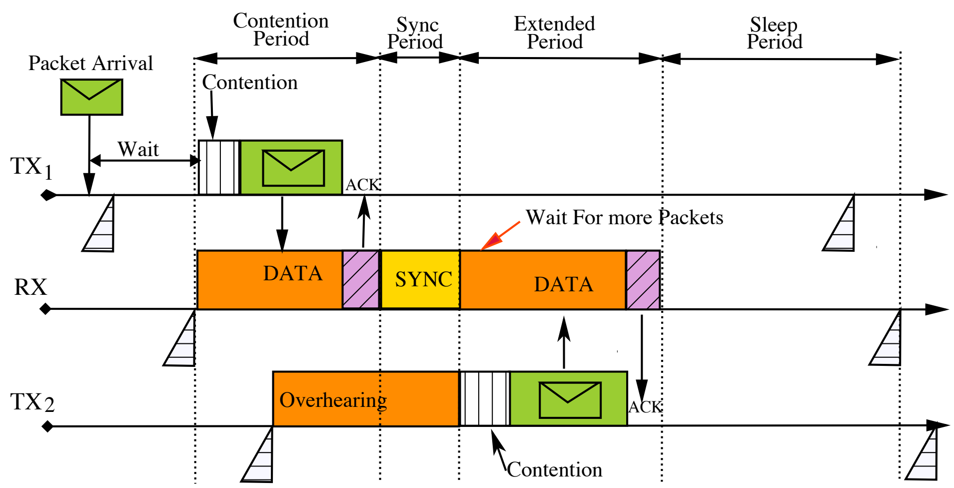

Fig. 11. ADCA-MAC's Extended period.

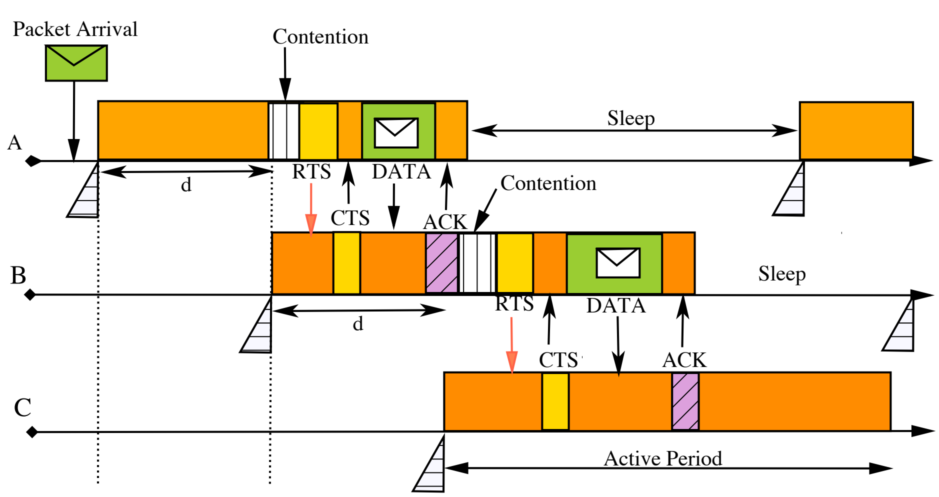

Fig. 12. LL-MAC's staggered wake-up.

to *ASYNC* packet, which is the number of hops from the node itself to the sink. When a node receives the *ASYNC* at startup stage, it will compare the hop information with its own. If the neighbor's hop information is bigger, it will subtract *d* from the neighbor's wake-up time. If the neighbor's hop information is smaller, it will add *d*, and if the neighbor's hop information is the same as its hop-count, it will follow the received schedule. LL-MAC support only converge-cast traffic (Second traffic pattern of section III), where nodes in WSN report their sensed data- either periodically or triggered by an event- to the sink along a spanning tree resulting in a so called many-to-one or converge-cast communication pattern. This makes the protocol suffer from flexibility to traffic patterns which limits its utilization. The expected delay of LL-MAC is given by Eq. (13):

$$\begin{aligned} OneHopDelay_{LL-MAC} &= t_{Sleep} + t_{Contention} + t_{RTS} \\ &\quad + t_{CTS} + t_{Packet} + t_{Ack} \end{aligned} \tag{13}$$

t_{Sleep}: time from event occurrence until the active time.

$$\begin{aligned} &EndToEndDelay_{LL-MAC} = (t_{Sleep} + t_{Contention} + t_{RTS} \\ &+ t_{CTS} + t_{Packet} + t_{Ack})^{1^{st}}_{packet} + \textstyle\sum_{i=2}^{n-1}(t_{Contention} \\ &+ t_{RTS} + t_{CTS} + t_{Packet} + t_{Ack})^{i^{th}}_{packet} = t_{Sleep} + nd \end{aligned} \tag{14}$$

where $d = t_{Contention} + t_{RTS} + t_{CTS} + t_{Packet} + t_{Ack}$. LL-MAC assumes a light traffic load and no latency caused by data transmission collision between nodes of the same level. Furthermore, LL-MAC does not consider clock drift, where two nodes may infinitely miss each other, when the clock drift θ becomes more than d. These two factors have direct impact on delay degradation.

A New-MAC based on 802.15.4 physical layer has been proposed in [99]. All nodes process setup stage during which schedule frames in the neighborhood are collected. If a node receives a schedule frame, it chooses a time window that will not overlap with the already scheduled ones and starts broadcasting its own schedule frame which contain its *id*, its *period* and its *oscillator's* drift. A node that receives a schedule frame, stores its information locally and consequently learns the consecutive moments that the corresponding neighbor node will be able to receive data. A node periodically wakes-up and broadcasts a schedule frame, then switch to listen state for potential data. If no channel activity sensed after T_L, the node goes-back to sleep. Otherwise, the reception procedure occurs, which includes *ACK*. After that, the node continues to listen for t_b time period for further possible transmissions before it returns to sleep. The proposed scheme has the same problem as LL-MAC, since it does not implement the preamble for compensating clock drift, where the sender can miss the receiver when the drift becomes longer than T_L period in the case where the time separating two event occurrences is long enough, which is frequent in event-driven WSN. This affects the end-to-end event delay.

AS-MAC [100] copes with the clock drift problem of LL-MAC and New-MAC. It is composed of initialization phase, periodic listening, and sleep phase. In the first phase, each node builds the neighbor table in which it stores neighbors' scheduling information, e.g., I_{wakeup}, I_{hello} and offset of the periodic wake-up offset (WO), and it chooses then announces its own and unique wakeup offset. At start-up, node listens to the channel for at least I_{hello} period to guarantee that it receives the Hello packets from all its neighbors. Then it sets its WO at the half point of the longest interval among the neighbors' WO. On Fig. 13, $WO_C = \frac{WO_A + WO_B}{2}$.

After that, the node starts the periodic listening and sleep phase setting wake-up interval, I_{wakeup}. A node performs LPL every I_{wakeup} to receive an incoming packet. If the channel is busy, the node receives the incoming packet. If the wake-up time of the node is also the *Hello* time, the node receives the packet after sending a *Hello* packet. When a node has a packet to send, it waits in sleep state until the receiver is scheduled to wake up, and it can predict the remaining time, t_{remain}, from the current time to the upcoming wake-up time of the receiver: If the wake-up time of the receiver is the *Hello* time, it receives the *Hello* packet and then sends the packet. Otherwise, it directly sends the packet with the preamble compensating clock drift (T_{G1} and T_{G2} guard time on Fig. 14). The sender performs collision avoidance backoff. If TX_2 loses the contention, it postpones its transmission to the RX's next wake-up time. With AS-MAC we have:

$$t_{remain} = I_{wakeup(i)} - (t_c - WO_i)\% I_{wakeup(i)} \tag{15}$$

where i is the ID of the receiver, and t_c is the current time. t_{remain} is a period from the current time to the upcoming wakeup time of the receiver. Let t_G be a guard time. We have $t_G = 2C_{drift}(t_c - WO)$, where C_{drift} is the maximum clock drift. The AS-MAC's delay is comparable to that of WiseMAC as they use the same technique and the expected AS-MAC delay is given by Eq. (16):

$$\begin{aligned} OneHopDelay_{AS-MAC} &= t_{Waiting} + t_G \\ &\quad + t_{Contention} + t_{Packet} + t_{Ack} \end{aligned} \tag{16}$$

$t_{Waiting}$: waiting time until the receiver wake-up.
$t_{Waiting} = t_{remain} - t_G$.
t_G: guard time used as a preamble to compensate the clock drift.

$$\begin{aligned} EndToEndDelay_{AS-MAC} &= \textstyle\sum_{i=1}^{n}(t_{Waiting} + t_G \\ &\quad + t_{Contention} + t_{Packet} + t_{Ack})_{i^{th}_{Hop}} \end{aligned} \tag{17}$$

AMAC [101] referrers to quorum-based asynchronous structure, realized over $PMAC's$ operation concept [102]. The idea is that the wake-up time is chosen to be partly overlapping with the neighboring nodes, so that the packet can be "heard". Under the quorum design, sensors entering sleep mode must wake-up periodically to send out beacon, and asynchronously they have to maintain certain wake-up time in the schedule matrix, so that the sender will have opportunities to send *DATA*. The solution can be interesting; however the main focus of this scheme is the maximization of energy saving, and it does not consider delay optimizing in its conception. Although these protocols set the active time to be partly overlapped with that of the next hop nodes in order to ensure quick packet forwarding , this may cause more contention from simultaneous transmitters and thus collisions which in turn affect both energy and delay efficiency in the network. Techniques such as minimizing contention and

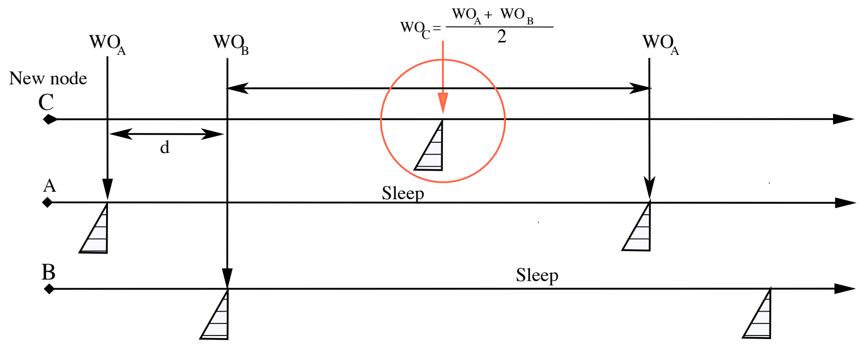

Fig. 13. AS-MAC Wakeup Schedule Setting at the initialization phase.

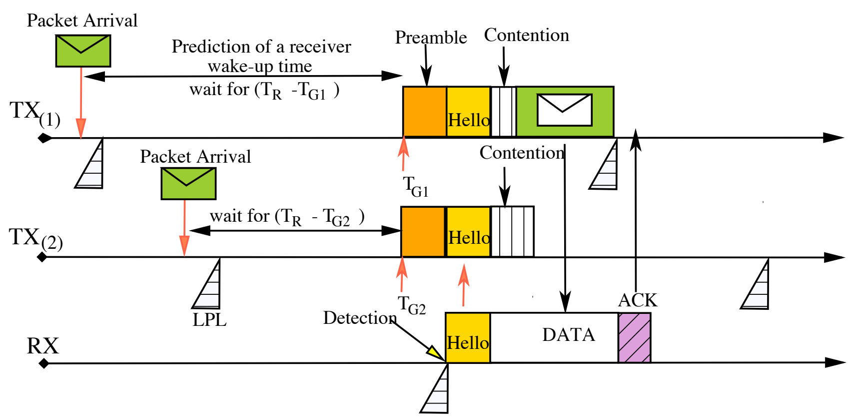

Fig. 14. AS-MAC Guard Time Preamble.

collisions among transmitters was introduces to improve the one-hop delay of a node.

VII. Collision Resolution Protocols

CSMA-based protocols such as 802.11 suffer from the delay caused by binary exponential backoff (BEB), which may take place when several sensors detect an event and attempt to report it simultaneously. Furthermore, the probability of collision still increases when using the uniform distribution in picking access slots, as done by B-MAC, S-MAC, and MACAW. This class of solutions tries to enhance the collision resolution procedure in order to minimize the one-hop delay.

Sift [103] focuses on the collision-free transmission of the first reports of some event between N neighboring nodes. It uses a small and fixed CW (contention window) interval, without binary exponential backoff (BEB). As shown in Fig. 15, transmissions from TX_1, TX_2 and TX_3 to the same receiver RX can occur without collisions using different slot pickup window for each node. The key difference between Sift and the previous CSMA-based MAC protocols is that the probability of picking a slot in this interval is not uniform. Thus, a carefully-chosen fixed, CW, and a geometric probability distribution, *Pr*, are applied.

$$Pr = \frac{1-\alpha^{cw}}{(1-\alpha)\alpha^{cw}} \times \alpha^{-r} \quad \text{for } r = 1, .., cw \tag{18}$$

where $0 < \alpha < 1$ is an application-related parameter. For these values of α, Pr increases exponentially with r, so later slots have higher probability.

$$\alpha = N_1^{\frac{1}{1-cw}} = 256^{\frac{1}{1-cw}} = 0,818 \tag{19}$$

N_1 is a fixed parameter that defines the maximum population size Sift is designed for. Empirical studies show that Sift delay degrades for $N_1 > 256$ [103]. For the choice of r, each node determines r as a slot to pick by a decision procedure within CW stages, which satisfies Eq. (18).

In Sift, if a node hears a report packet when two or more neighbors detect the same event, then it cancels its packet transmission. This leads to minimizing a collision probability and increases event reporting delay. However, Sift does not consider duty-cycling, the overhearing problem, idle listening and their impact on delay degradation. Furthermore, the choice of CW's fixed size is not argued.

The estimated number of contenders (ENCO) method [104] was proposed for event-driven WSNs based on the mean coverage degree, denoted in what follows by C. CW sizes that optimize energy and delay can be found using Eq. (20). The knowledge of the number of contending nodes used in the equation, N, may be gotten by approximation method of C. If the deployment is uniform, d is the density of the deployment area, and R_s is the sensing range of the sensors, then $C = \Pi R_s 2d$.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

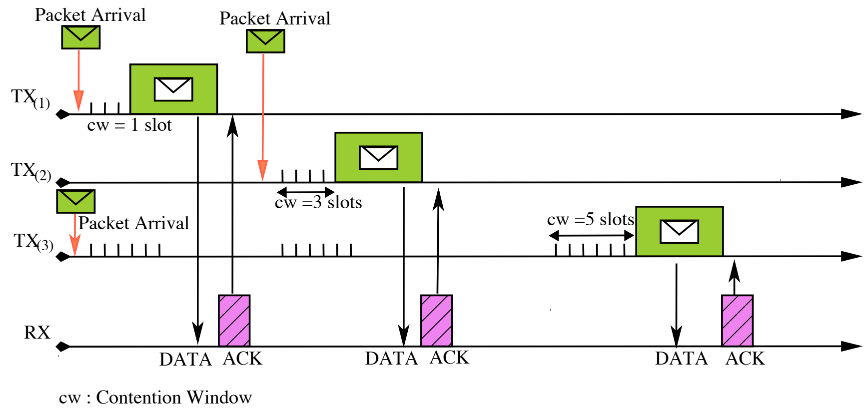

Fig. 15. Sift-MAC's Collision Free Transmissions.

The channel access delay is defined as the expected duration between the beginnings of the contention to a successful medium access, denoted by Ω. It can be divided into two phases:

1) The expected time spent for collision and retrials till the beginning of the collision-free slot selection "Λ".
2) And the expected carrier sense duration within the successful contention, from the beginning of the contention window till the first occupied slot "Γ".

The equations of both components are derived in terms of the contention window size W and the number of contending nodes N. At each contention, N nodes independently and randomly select a slot from the uniform slot distribution $1, ..., W$. Let the random variable Ψ represent the first occupied slot number, regardless whether the slot selection results in collision or not. Ω delay is defined in [104] as:

$$\begin{aligned}\Omega = \Lambda + \Gamma = (\tfrac{1}{\xi} - 1) \times (t_c \\ + \sum_{\Psi=1}^{w} \frac{1+\sum_{m=2}^{N-1}((\frac{N}{m})(w-\Psi)^{N-m})}{w^N - N\sum_{f=1}^{N-1}(w-f)^{N-1}} \times (\Psi - 1)t_s) \\ + \sum_{\Psi=1}^{w}(\Psi - 1)\frac{(w-\Psi)^{N-1}}{\sum_{f=1}^{w}(w-f)^{N-1}} t_s\end{aligned} \quad (20)$$

Where t_s is one slot duration, ξ is the collision probability at the contention of N nodes, t_c is the duration from the collision detection to the beginning of the new contention (collision timeout), and finally m represents the number of nodes that selected the slot Ψ.

Note that for all the above reviewed classes of protocols, the communications are sender initiated, i.e., the send is responsible for waking-up the receiver to exchange data packets using different kind of wakeup preambles. Receiver initiated communications are explored thus far in the literature.

VIII. Receiver Initiated Protocols

In this scheme, nodes' schedules are set independently, but the receiver is the responsible for communication initiation by broadcasting beacon at the time of its wake-up. So there is no need to send wakeup preamble saving thus energy at the sender side. The typical protocol that has used this technique is RI-MAC protocol introduced by Sun et al. [105].

RI-MAC [105] tries to reduce the amount of time a pair of nodes occupy the medium by preambles before they reach a rendezvous time for data exchange, at the aim of reducing the global network delivery delay. Each node periodically wakes-up and broadcasts a beacon. Refer to Fig. 16, when node $TX_{(1)}$ wants to send a $DATA$ frame to node RX, it stays silently active and starts $DATA$ transmission upon receiving a beacon from RX, which will be acknowledged by RX with another beacon. Note that this ACK beacon's role is twofold: first, it acknowledges the correct receipt of the sent $DATA$ frame, and second, it invites a new $DATA$ frame transmission from $TX_{(2)}$ to the same receiver. A-MAC [106] improves RI-MAC in term of rapid turning on/off the radio after the probing beacon. In A-MAC, nodes with data pending use a small *auto-ACK* packet in response to the receiver's wakeup beacon, which enables it to quickly make the decision whether to keep the radio on or to turn it off. But this come at the cost of adding some delay due to the *auto-ACK* signal. The expected Delay of RI-MAC is given by Eq. (21):

$$\begin{aligned}OneHopDelay_{RI-MAC} = t_{Waiting} + t_{Beacon} + t_{Contention} \\ + t_{Packet} + t_{Beacon\_Ack}\end{aligned} \quad (21)$$

$t_{Waiting}$: waiting time until the reception of the destination Beacon.

$$\begin{aligned}EndToEndDelay_{RI-MAC} = \sum_{i=1}^{n}(t_{Waiting} + t_{Beacon} \\ + t_{Contention} + t_{Packet} + t_{Beacon\_Ack})i^{th}_{Hop}\end{aligned} \quad (22)$$

In RI-MAC, the maximum data transmission delay used for collision detection or for return to sleep is not fixed, and it is not the same for every node. This may lead to missing upcoming $DATA$. Thus, RI-MAC uses no contention window (CW) for the basic (first) beacon, which can cause collision in intermediate nodes, and the binary exponential backoff (BEB) can degrade the end-to-end delay. Furthermore, if no sender has data to send then there is no need to occupy the channel by beacons. RICER [83] is another protocol that follows the same scheme, and is very similar to RI-MAC.

The Breath protocol [107] is a cross-layer solution which assumes that data and beacon packets are transmitted at two fixed disjoint frequencies. Nodes are grouped into clusters and their size changes according to the traffic load. A node sends a data packet to a node randomly selected in a forwarding region, which is located in the direction toward the sink node.

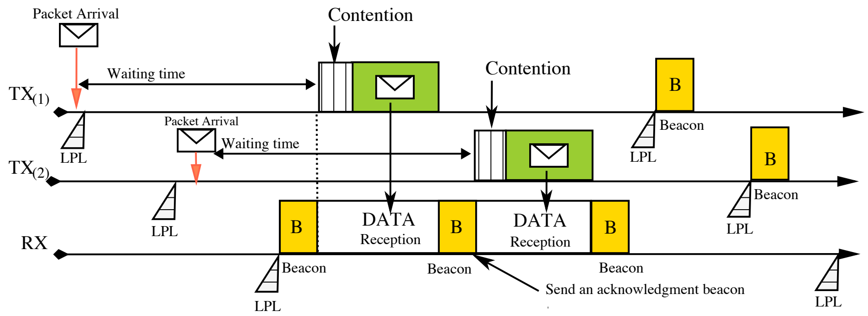

Fig. 16. RI-MAC Receiver Initiated Communication technique.

Nodes in the forwarding region send beacon messages to say that they are available to receive data packets. The protocol is probabilistic and does not implement any acknowledgement or retransmission scheme. Each node, either transmitter or receiver, does not stay in an active state all the time, but goes to sleep for a random amount of time, which depends on the traffic conditions. A node in the Active-State starts a waiting timer of a fixed duration. If the node receives the first beacon coming from a node in the forwarding region within the waiting time, it uses $CSMA$ to send data. Otherwise, it goes to the Sleep-State. Note that the time to wait before the first wake-up of a node in the next cluster is an exponentially distributed random (Poisson) variable, and the time to wait after collision is an exponentially distributed random backoff time, which significantly affects the one-hop delay.

Receiver-initiated is a reverse approach of Preamble sampling; Instead of listening periodically, beacons are sent at regular intervals, indicating that the node is ready to subsequently receive data. This avoids sending a long wakeup preamble (the sender keeps receiving instead of transmitting a full-length preamble), and it saves more energy at the sender side. But this may add control packet overhead in lightly loaded networks, as the receiver makes invitations periodically [38]. The drawback is that beacons interfere with ordinary traffic, and with one other. The one-hop delay depends on the receiver wake-up interval, which is determined by the duty cycling rate factor. Any-casting technique was proposed to cope with the unknown receiver wake-up time. By anticipating the forwarding of the packet to the first awake neighbor, the end-to-end delay can be improved.

IX. Anticipating Protocols

In this approach, nodes' schedules are set independently, and a node tries to minimize data delivery delay by anticipating its transmissions. To ensure this, a node sends its packet to the first waked-up node among its neighbors. The following protocols are some examples:

HES-MAC protocol [108] applies periodic sleep approach, where the time is divided into frames with a fixed length, composed of wake-up period and sleep period. The sleep period can be much longer than the listen period in this protocol. The periodic sleeping in HES-MAC is similar to that of S-MAC, but HES-MAC does not rely on synchronization. It uses 802.11 RTS/CTS with BEB. If a node does not receive the CTS for the third time, it can deliver the message to one of its one-hop neighbors, using RTS/CTS exchange with BEB, and it requests the selected neighbor (called a bridge) to forward the message to the destination node. The source node selects the bridge node based on the first correctly received $BrdCTS$ on the response to $BrdRTS$ of the source TX on Fig. 17). HES-MAC does not send ACK to the actual source. Furthermore, the Source_Next-hop delay can be more than the Source_Bridge_Next-hop delay. Due to the use of three RTS tries, CTS and the $AwakeBeacon$, in addition to the queuing time, the end_to_end delay may considerably increase. The expected Delay of HES-MAC protocol is given by Eq. (23):

$$\begin{aligned} OneHopDelay_{HES-MAC} = (t_{Contention} + t_{RTS} + t_{CTS} + t_{Packet} \\ + t_{Ack}) \times P_{Direct\_Success} + (t_{Contention} + 3 \times t_{RTS} \\ + t_{BrdRTS} + t_{BrdCTS} + t_{Packet} + t_{Ack} + t_{time\_to\_wakeup} \\ + t_{AwakeBeacon} + t_{Contention} + t_{RTS} + t_{CTS} \\ + t_{Packet} + t_{Ack}) \times (1 - P_{Direct\_Success}) \times P_{Bridge\_Success} \end{aligned} \tag{23}$$

$K \in 1, .., 3$, is the number of RTS tries.
$P_{Direct\_Success}$: the probability of success when trying to send to the destination neighbor directly.
$P_{Bridge\_Success}$: the probability that the neighbor accepts to be a bridge.

CMAC [109] is yet another anticipating protocol in which the sender node transmits data to the first and furthest next-hop node toward the sink to minimize as mush as the forwarding delay using GeRAF [110] geographical random forwarding technique.

In [111], the optimal sleep-wake scheduling and anycast forwarding policies are developed for some wake-up patterns. First, with Poisson wake-up pattern, and later with periodic wake-up patterns [112]. The asynchronous sleep-wake scheduling consists of two phases: the configuration phase during which the nodes determine their optimal packet-forwarding and sleep-wake scheduling policies f^*, followed by the operation phase. From Fig. 18 as soon as a node, Tx, is ready to transmit a packet, it sends a beacon signal of duration t_B, and ID signal of duration t_C, and then listens for acknowledgements (CTS) for duration t_A. The sending node repeats this sequence until it hears an acknowledgement. The ID signal contains the identity of the sending node and

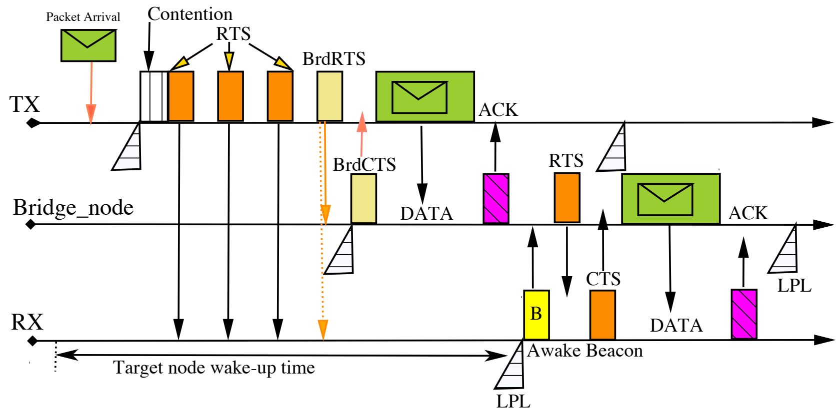

Fig. 17. HES-MAC: Anticipated Sending using a Bridge node.

the sequence number of the last beacon signal. When a node wakes-up and senses the h^{th} beacon signal, it will stay awake to decode the following ID signal. Then, it responds with CTS. If the awaked node is in the forwarding set and it fulfills the delay-optimal forwarding policy f^*, it chooses minimum delay between sending the packet to this neighbor and waiting to the next awaked node as expressed by Eq. (24):

$$d^{(h)}(x_h) = min(d^{(h)}_{wait}(x_h), d^{(h)}_{TX}(x_h)) \tag{24}$$

$d^{(h)}_{wait}(x_h)$ is the expected delay incurred if it chooses to wait for the next neighbor to wake-up, and $d^{(h)}_{TX}(x_h)$ is the expected delay if it chooses to send the packet at h^{th} beacon signal.

The Anycasting policy is then formulated as a dynamic programming problem and resolved by a value iteration algorithm, which attempts to find the optimal solution. However, defining the node-to-sink delay in the configuration phase is not the best choice, because the one-hop delay changes dynamically and the collision between awaked nodes can affect the delay. Furthermore, this scheme is not flexible for dynamic topology change, as the configuration phase must be executed for each newly joined node and after node failure. Additionally, concurrent sending and waked-up nodes can engender collision, which affects the overall network delay.

X. Review Summary and Futur Directions

In this section, we summarize key features of the reviewed protocols and recap how these features affect the transmission delay of the delivered data. Suitability of these protocols for delay-sensitive applications is discussed, as well as remaining research problems and future directions.

A. Review Summary and Discussions

The real need for power management has been a driving force behind the development of many WSN-specific medium access control protocols. Asynchronous protocols are very appealing for low data rate multi-hop wireless sensor network (WSN). Each protocol strikes a different balance between delay performance and energy consumption; all claiming to be more efficient than the canonical B-MAC protocol. But they are evaluated with different workloads, simulation environments, and hardware platforms, making it hard to assess the true benefits of the individual MAC protocols. In asynchronous contention-based MAC protocols, the sleep delay is a serious drawback that increases communication latency. This delay is the time that a node waits until the wake-up of the forwarder node on every hop. Many improvements have been proposed to minimize the sleep delay. following our taxonomy, asynchronous MAC protocols can be classified into six different categories. Fig. 19 depicts timeline development of key protocols from each category.

Static wake-up preamble protocols are simple to be implemented on most low-cost microcontrollers. However, they lack flexibility to traffic load fluctuation, and they also introduce channel overhead due to the long wake-up preamble size (e.g. B-MAC). This can significantly affect the transmission delay of a node. As expected, the message transfer time is prolonged by a long preamble spanning the complete polling period T_w. This represents the one-hop delay of B-MAC, where T_w is the duration of periodic wake-up time. B-MAC does not achieve any delay decrease. Whereas, the early ACK – introduced by X-MAC as a notification to interrupt the strobe small preambles– contributes in reducing the one-hop delay by a factor of two. Hence, the average one-hop delay of X-MAC is $T_w/2$, but this protocol still suffers from channel overhead. Static wake-up preamble protocols are suitable for very low rate data applications, but they may fail to meet requirements of delay-sensitive applications. To cope with channel overhead, adaptive wake-up preamble protocols use a preamble with a size proportional to the data rate, (e.g., WiseMAC). This keeps the channel empty as long as possible, and it gives other concurrent transmissions the opportunity to forward packets without additional cycles. WiseMAC determines the starting point of the preamble based on the estimated wakeup time of the receiving node, subtracting half the dynamically adapted guard-time. For a single contender based network, the latency of WiseMAC is similar to that of X-MAC. On average, the sending node has to wait for, $T_w/2$, before starting to transmit the wakeup preamble and a data packet. As illustrated in Section V, a node in adaptive wake-up preamble protocols must stay waiting in sleep mode for its neighbor's wake-up time to forward data packets. This sleep-time may result in high latency, especially for multi-hop networks. Receiver-initiation or beacon-based approach is another way to pro-

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

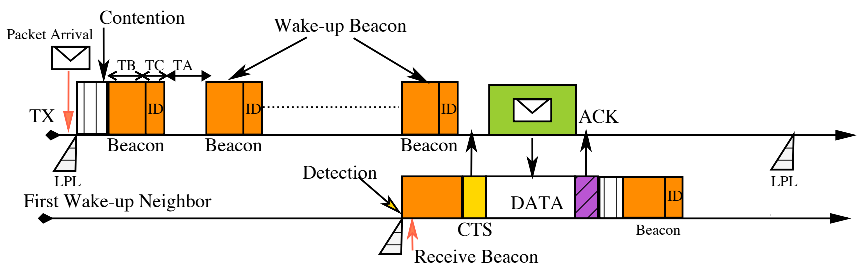

Fig. 18. Delay-Aware Anycasting strategy.

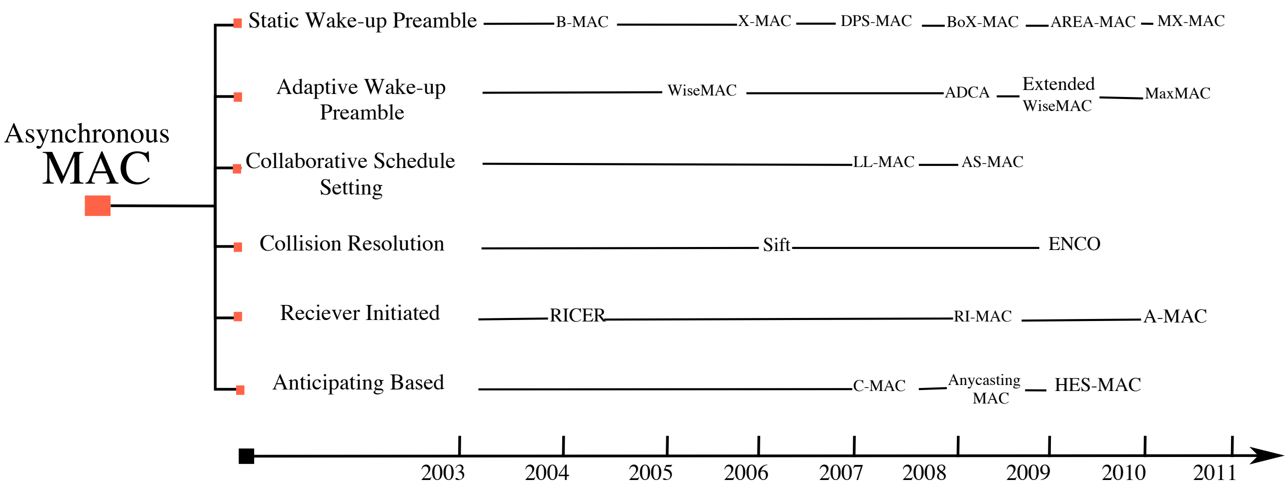

Fig. 19. Asynchronous MAC Protocols' Timeline

TABLE III
ASYNCHRONOUS MAC PROTOCOLS SUMMARY

Category	Protocol(s)	Delay Decrease	Cross-layer Support	Prioritized MAC	Delay enhancing mechanism	Traffic Pattern	Traffic Load
Static Wakeup Preamble	B-MAC[26], X-MAC[52], EA-ALPL[77], SEESAW[78], DPS-MAC[79], Patterned Preamble[80], AREAMAC[81], CSMA-MPS[82], TICER[83], MH-MAC[84],STEM[85], BoX-MAC[86],SpeckMAC-D[87], MX-MAC[88], and RA-MAC[89]	No (e.g B-MAC), One-hop (e.g X-MAC)	No	No	No (e.g B-MAC), strobe wake-up preamble (e.g X-MAC)	Bidirectional (Random topology)	Light
Adaptive Wakeup Preamble	WiseMAC[45], MaxMAC[92],AS-MAC[93], ADCA[94], SP[95], SyncWUF [96], and RATE EST[97]	One-hop (e.g WiseMAC)	No	No	Learn neighbors schedule and use minimized wakeup preamble	Bidirectional (Random topology)	Moderate
Collaborative Schedule Setting	LL-MAC[98], New-MAC[99], AS-MAC[100], and AMAC[101]	Probabilistic End-to-End Decrease/SRT	Routing	No	Set its wakeup according to $neighbors'$ schedule for minimizing delay	Unidirectional asymmetric (Oriented Graph, Tree topology)	Moderate
Collision Resolution	Sift[103], and ENCO[104]	One-hop (e.g Sift)	No	No	Enhance Backoff mechanism to minimizing channel access delay	Bidirectional (Random topology)	Moderate
Receiver Initiated	RICER[83], RI-MAC[105], A-MAC[106], and Breath[107]	One-hop (e.g RI-MAC)	No	No	Use Wakeup beacon to minimize channel occupation time	Bidirectional (Random topology)	Moderate
Anticipating	HES-MAC [108], CMAC [109], and Optimal Anycasting Technique [111], [112]	Probabilistic End-to-End Decrease/SRT (e.g HES-MAC)	Routing	No	Send packet to the first waked-up neighbor toward the sink	Unidirectional asymmetric (Oriented Graph, Tree topology)	Moderate

vide adaptability (e.g. RI-MAC), where the communication is initiated by the receiver instead of the sender. This avoids sending a long wakeup preamble (the sender has to listen for the receiver's beacon) and shortens transmission time compared to static wake-up preamble protocols like B-MAC. A serious drawback is possible interference between beacons, and between beacons and ordinary traffic. Although Adaptive wake-up preamble and beacon-based protocols reduce channel occupation, latency may be significant when nodes stay long in sleep state waiting for their neighboring nodes to wake-up. The two classes of protocols may be effective in very small networks with bounded response-time. However, They cannot be applicable to medium and large scale networks with time-critical requirements. Collaborative schedule setting protocols –such as LL-MAC– try to tackle the waiting time problem by setting the wake-up time of neighboring nodes in staggered mode, so that a node's active time is partly overlapping with its predecessor and its successor active times. To ensure this, routing information about next-hop nodes in the route toward the sink is needed, as well as periodic schedule exchange, to overcome clock drifts. However, assumption on traffic pattern to be converge-cast is generally made. Furthermore, setting active periods in a cascading way lead to more contention and collisions, which is the main source of energy and delay degradation. Collaborative schedule setting protocols can achieve probabilistic end-to-end delay decrease. Collision-resolution protocols aim to minimize the impact of collisions on the transmission delay. For instance, protocols like Sift propose a carrier sensing function that provides collision-free access to the first N data packets that are generated as soon as an event is detected. These solutions might be interesting when combined with other delay reduction techniques of other categories. Delay reduction can be achieved using any-casting technique, where nodes anticipate their transmission and send packets to the first waked-up neighbor in the forwarding region toward the sink, e.g., HES-MAC. However, to enable a node choosing between sending out a packet to the awaked neighbor and waiting for the destination node to wake-up requires routing information and routes' delay estimation to be available. Moreover, protocols of this class rely on converge-cast communications traffic in their conception. As a result, they are suboptimal and do not meet all traffic pattern requirements in terms of delay guarantee, but they provide a probabilistic end-to-end delay decrease compared with B-MAC.

As shown in Table III that summarizes all the reviewed protocols of each MAC category, as well as the mechanism affecting the delay, none of the reviewed protocols is basically designed to support data prioritization through service differentiation or other mechanisms. Aad and Castelluccia [113] present the first three service differentiation schemes for IEEE 802.11 that can be applied to WSNs. They have introduced DIFS (Distributed Inter Frame Space) that enables to support data prioritization. Prioritized MAC enables to satisfy delay requirements of each individual data packet in order to meet its deadline. Such a mechanism is important when designing real-time system embedded in delay-sensitive wireless sensor networks.

B. Open Research Directions

Application fields of the WSNs are growing rapidly as the capabilities of the tiny sensor devices improve. Most of WSN future applications will require timely delivered data. Moreover, diversity of the applications yields heterogeneous WSNs composed of multimodal sensor nodes, which provide more than one functionality by delivering multiple types of traffic. Therefore, novel MAC protocols that have the ability to fulfill the application requirements in terms of delay guarantee for each kind of traffic are required.

Whilst many power-efficient MAC protocols have been proposed, numerous issues related to minimizing transmission delay are still wide open. One critical issue with the asynchronous wake-up protocols is how to define a rendezvous mechanism that permits a sender reaching its targeted receiver with a minimum waiting time. One interesting way to achieve this is to employ adaptive preamble combined with collaborative scheduling to stagger a wake-up time of a forwarder node in a wave-like mode, along with building a tree topology starting from the sink as a root. Consequently, the wake-up schedule of each node would be determined by the level of the node in the tree. However, book-keeping of forwarders' wake-up offset is necessary. This can be done using some relaxed synchronization. Another important issue with the existing wake-up preamble protocols is channel overhearing, i.e., high idle listening. Nodes use adaptive-listening (overhearing) to accelerate message delivery, but at the cost of more energy consumption. It is still challenging to cope with the sleep delay problem while ensuring adaptability to traffic fluctuation. Combining techniques of asynchronous schemes– that provide energy-efficiency– with those of synchronous schemes that minimize the latency might be a promising approach. Cross-layer design is also promising. For instance, it is possible to use a joined routing and MAC to build a TDMA tree structure with relaxed synchronization and hybrid access strategy. Most of the current solutions only provide soft real-time assurance. This is due to the absence of any synchronization and collaboration between nodes, and the lossy nature of the under-laying platform over which the application runs. Real-time properties should be provided by the underlying operating system, before the high-level applications can implement it. A lot of work has to be done before achieving reliable MAC protocols, which must be effective in energy conservation while providing delay guarantee.

XI. Conclusion

Wireless Sensor Networks (WSNs) have progressively changed the way to monitor the environmental and industrial phenomena over the last two decades. Since these networks use wireless channels, the medium access control is of pivotal importance. Asynchronous contention-based medium access control (MAC) protocols are widely used for relaxed traffic load conditions in multi-hop WSN, notably for periodic, on-demand, and event-driven applications. Many energy and delay constrained application scenarios can be envisaged for WSN. Energy-efficiency is generally achieved by duty-cycling the nodes' radios, which makes nodes sleep most of the time and wake-up just when it is necessary. However, low duty-cycling usually comes on the cost of the transmission delay increase, resulting in a trade-off between energy-consumption and network performance in terms of latency. The state-of-art literature includes a huge number of asynchronous energy-efficient MAC protocols, which trade energy saving with delay reduction. However, each protocol has its own specific mechanism that deals with this trade-off. The concept of WSN MAC protocol timeliness has been investigated in this paper, where a novel taxonomy has been introduced. Compared to the traditional classification based on medium access principles, or on the issues targeted by protocols, the one proposed in this paper is based on the mechanisms introduced by the solutions that affect the transmission delay. Well-known energy-efficient asynchronous medium access control protocols have been discussed from the latency point of view. Six different categories of asynchronous MAC protocols have been reported; static wake-up preamble (Section IV), adaptive wake-up preamble (Section V), collaborative schedule setting protocols (Section VI), collision resolution protocols (Section VII), receiver-initiated (Section VIII), and last but not least anticipation-based protocols (Section IX). Table III summarizes the presented protocols on the perspective of this taxonomy. Our review indicates that some of these protocols can ensure end-to-end delay decrease, but none of them can provide delay guarantee for time-constrained applications. Consequently, a huge effort is needed in order to enable successful deployment of WSNs for delay-sensitive applications using asynchronous MAC.

REFERENCES

[1] CC2420 Single-Chip 2.4 GHz RF Transceiver, pp. 13, Retrieved from Texas Instruments. URL: *http://focus.ti.com/lit/ds/symlink/cc2420.pdf*, Mar. 2010.

[2] 8-bit Microcontroller with 128K Bytes In-System Programmable Flash, ATmega128, pp. 340, Retrieved from Atmel Corporation website. URL: *http://www.atmel.com/*.

[3] T. Wark, P. Corke, P. Sikka, L. Klingbeil, Y. Guo, C. Crossman, P. Valencia, D. Swain, and G. Bishop-Hurley, "Transforming agriculture through pervasive wireless sensor networks," *IEEE Pervasive Computing*, 2007, Col. 6, No. 2, pp. 5057.

[4] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," *in: WSNA02: Proc. 1st ACM International Workshop on Wireless Sensor Networks and Applications*, ACM, New York, NY, USA, 2002, pp. 8897.

[5] A. Milenkovic, C. Otto, and E. Jovanov, "Wireless sensor networks for personal health monitoring: issues and an implementation," *Computer Communications*, Feb. 11, 2006, Vol. 29, No. 1314, pp. 25212533.

[6] V. Gungor and G. Hancke, "Industrial wireless sensor networks: challenges, design principles, and technical approaches," *IEEE Trans. Ind. Electron.*, 2009, Vol. 56, No. 10, pp. 42584265.

[7] I. Akyildiz, T. Melodia, and K. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, 2007, Vol. 51, No. 4, pp. 921960.

[8] S. Soro, and W. Heinzelman, "A survey of visual sensor networks," *Advances in Multimedia*, 2009, pp. 122.

[9] M.A. Yigitel, O.D. Incel and C. Ersoy, "QoS-Aware MAC Protocols for Wireless Sensor Networks: A Survey," *Computer Networks (Elsevier)*, June 2011, Vol. 55, No. 8, pp. 1982-2004.

[10] K. Akkaya, M. F. Younis: "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, 2005, Vol. 3, No 3, pp 325-349.

[11] S. Coleri Ergen and P. Varaiya, "Energy efficient routing with delay guarantee for sensor networks," *ACM Wireless Networks*,Oct. 2007, Vol. 13,Issue 5, pp. 679-690.

[12] T. He, J.A. Stankovic, C. Lu and T. Abdelzaher, "SPEED: a stateless protocol for real-time communication in sensor networks," *In International Conference on Distributed Computing Systems (ICDCS '03)*, Washington, DC, USA, May 2003, No. 46, pp. 46-55.

[13] E. Felemban, C.G. Lee and E. Ekici, "MMSPEED: Multipath multi-speed protocol for QoS guarantee of reliability and timeliness in wireless sensor network," *IEEE Trans. Mobile Comput.*, 2006, Vol. 5, No. 6, pp. 738-754.

[14] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic and T. Abdelzaher, "RPAR: Real-time power-aware routing in sensor network," *in the Proc. 14th IEEE International Workshop on Quality of Service (IWQoS'06)*, New Haven, CT, June 2006, pp. 83-92.

[15] K. Akkaya, and M. Younis, "An energy-aware QoS routing protocol for wireless sensor networks," *Proc. IEEE Workshop on Mobile and Wireless Net works (MWN'03)*, Providence, RI, May 2003, pp. 710-715.

[16] Y. Li, C. Shue, Y. Song, Z. Wang and T. Sun, "Enhancing Real-Time Delivery in Wireless Sensor Networks With Two-Hop Information," *IEEE Trans. Ind. Informat. (TII)*, 2009, Vol. 5, No. 2, pp. 113-122.

[17] P. Rezayat, M. Mahdavi, M. Ghasemzadeh, M. A. Sarram, "A Novel Real-Time Power Aware Routing Protocol in Wireless Sensor Networks," *In International Journal of Computer Science and Network Security (IJCSNS)*, April 2010, Vol. 10, No. 4.

[18] A. Razzaque, M. M. Alam, M. Or-Rashid and C. S. Hong, "Multi-constrained QoS geographic routing for heterogeneous traffic in sensor networks," *IEICE Trans. Commun.*, 2008, Vol. 91 B, No. 8, pp. 589-2601.

[19] D. Djenouri and I. Balasingham, "Traffic-Differentiation-Based Modular QoS Localized Routing for Wireless Sensor Networks," *IEEE Trans. Mobile Computing*, June 2011, Vol. 10, No. 6, pp. 797-809.

[20] K. Zeng, K. Ren, W. Lou, and P. J. Moran, "Energy aware efficient geographic routing in lossy wireless sensor networks with environmental energy supply," *Wireless Networks*, 2009, Vol. 15, No. 01, pp. 3951.

[21] A. Mahapatra, K. Anand, and D. P. Agrawal, "QoS and energy aware routing for real-time traffic in wireless sensor networks," *Computer Communications*, 2006, Vol. 29, No. 4, pp. 437445.

[22] A. R. Sama and K. Akkaya: "Real-time routing for mobile sensor/actor networks," *33rd IEEE conference on Local Computer Networks LCN08*, Montreal, Canada, 2008, pp. 821-828.

[23] J. Chen, M. Diaz, L. Llopis, B. Rubio, J. M. Troya, "A survey on quality of service support in wireless sensor and actor networks: Requirements and challenges in the context of critical infrastructure protection," *Journal of Network and Computer Applications*, July 2011, Vol. 34, Issue 4, pp. 1225-1239.

[24] Z. An-dong, X. Tian-yin, C. Gui-hai, Y. Bao-liu and L. Sang-lu, "A Survey on Real-time Routing Protocols for Wireless Sensor Networks," *Second Chinese Conference on Sensor Networks (CWSN'08)*, 2008.

[25] C. LU, B. BLUM, T. Abdelzaher et al., "RAP: A Real-time Communication Architecture for Large-Scale Wireless Sensor Networks," *In the Proc. Eighth IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS02)*, 2002, pp. 55-66.

[26] J. Polastre, J. Hill and D. Culler, "Versatile low power media access for wireless sensor networks," *Proc. 2nd international conference on Embedded networked sensor systems (SensSys04)*, USA. November 03-05, 2004, pp 95-107.

[27] [Online]. Available: *http://www.wavenis-osa.org/*

[28] [Online]. Available: *http://www.knx.org/*

[29] IEEE Std 802.15.4-2006: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), IEEE Std., September 2006.

[30] HART Communication Foundation, "WirelessHART Technology," URL: *http://www.hartcomm.org/protocol/wihart/wireless_technology.html*, Dec. 2009.

[31] [Online]. Available: *http://www.ietf.org/html.charters/6lowpan-charter.html*

[32] [Online]. Available: *http://www.ietf.org/html.charters/roll-charter.html*

[33] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC Protocols for Wireless Sensor Networks: a Survey," *IEEE Commun. Mag.*, April 2006, Vol. 06, pp. 115121.

[34] B. Yahya and J. Ben-Othman, "Towards a classification of energy aware MAC protocols for wireless sensor networks," *Wireless Communications & Mobile Computing*, December 2009, Vol. 9, No. 12, pp.1572-1607.

[35] K. I. Kredo and P. Mohapatra, "Medium Access Control in Wireless Sensor Networks," *Computer Networks*, March 2007, Vol. 51, No. 51, pp. 961994.

[36] K. Langendoen. "Medium access control in wireless sensor networks," *Medium Access Control in Wireless Networks: Practice and Standards*, Nova Science Publishers Inc, 2008, Vol. 2, pp. 535-560.

[37] R. Kuntz, "Medium Access Control Facing the Dynamics of Wireless Sensor Networks," *PhD Thesis*, September 2010.

[38] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, “MAC Essentials for Wireless Sensor Networks,” *IEEE Commun. Surveys Tutorials*, 2010, Vol. 12, No. 2.

[39] C. Zhou and T. Zhu, “Thorough Analysis of MAC Protocols in Wireless Sensor Networks,” *4th IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM’08)*, Dublin, Oct. 2008, pp. 1-4.

[40] M. Ali, A. Bhm and M. Jonsson, “Wireless Sensor Networks for Surveillance Applications A Comparative Survey of MAC Protocols,” *in Proc. 4th International Conference on Wireless and Mobile Communications (ICWMC’08)*, Athens, USA, July 27 -Aug. 1, 2008, pp. 399-403.

[41] M. Miladi, T. Ezzedine and R. Bouallegue, “Latency of Energy Efficient MAC Protocols for Wireless Sensor Networks,” *Proc. international conference on Digital Telecommunications (ICDT’06)*, Cote d’Azur, France, August 29-31, 2006, pp 60-60.

[42] K. Langendoen and A. Meier, “Analyzing MAC Protocols for Low Data-Rate Applications,” *ACM Transactions on Sensor Networks*, ACM New York, USA, Vol. 7, Issue 1, Article 10, August 2010.

[43] V. Hoesel, L.F.W. and Havinga, P.J.M, “A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks: Reducing Preamble Transmissions and Transceiver State Switchess,” *In 1st International Workshop on Networked Sensing Systems (INSS’04)*, Tokio, Japan 2004.

[44] W. Ye, F. Silva and J. Heidemann, “Ultra-low duty cycle mac with scheduled channel polling,” *In Proc. 4th ACM International conference on Embedded networked sensor systems (SenSys’06)*, New York, USA, 2006. pp 321334.

[45] A. El-Hoiydi and J.-D. Decotignie, ‘Low Power Downlink MAC Protocols for Infrastructure Wireless Sensor Networks,” *Mobile Networks and Applications (MONET)*, ACM/Kluwer Academic Publishers, 2005, Vol 10, No 06, pp 675-690.

[46] G.P. Halkes and K.G. Langendoen, “Crankshaft: An Energy-Efficient MAC-Protocol For Dense Wireless Sensor Networks,” *Proc. 4th European Conference on Wireless Sensor Networks (EWSN’07)*, Springer 2007, pp. 228-224.

[47] P. Suriyachai, U. Roedig, and A. Scott, “A Survey of MAC Protocols for Mission-Critical Applications in Wireless Sensor Networks,” *IEEE Communications Surveys & Tutorials*, March 2011, Issue 99, pp 1-25.

[48] TinyOS 2.x. URL: *http://www.tinyos.net/*.

[49] J. Arfvidsson, E. Park, and P. Levis, “Lowering radio duty cycle through temperature compensated timing,” *Proc. 4th international conference on Embedded networked sensor systems*, 2006, pp. 409410.

[50] Wei Ye, John Heidemann, and Deborah Estrin, “Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks,” *IEEE/ACM Trans. Netw.*, June 2004, Vol. 12, No. 3.

[51] G. Lu, B. Krishnamachari and C. S. Raghavendra: “An adaptive energy-efficient and low-latency MAC for tree-based data gathering in sensor networks,” *Wireless Communications and Mobile Computing*, 2007, Vol. 7, No 7, pp. 863-875.

[52] M. Buettner, G. Yee, E. Anderson and R. Han, “X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks,” *Proc. 4th international conference on Embedded networked sensor systems (SenSys06)*, Boulder, Colorado, USA, November 13, 2006, pp. 307-320.

[53] S. C. Ergen and P. Varaiya, “PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks,” *IEEE Trans. Mob. Comput.* 2006, Vol. 5, No. 7, pp. 920-930.

[54] A. Sahoo and P. Baronia, “An Energy Efficient MAC in Wireless Sensor Networks to Provide Delay Guarantee,” *15th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN)*, Princeton, NJ, USA, June 2007.

[55] S. Munir, S. Lin, E. Hoque, S.M.S. Nirjon, J. A. Stankovic, and K. Whitehouse, “Addressing Burstiness for Reliable Communication and Latency Bound Generation in Wireless Sensor Networks,” *in Proc. 9th ACM/IEEE Int. Conf. Information Processing in Sensor Networks*, Stockholm, Sweden, 2010, pp. 303-314.

[56] P. Suriyachai, J. Brown, and U. Roedig, “Time-Critical Data Delivery in Wireless Sensor Networks,” *in Proc. 6th IEEE Int. Conf. Distributed Computing in Sensor Systems*, Santa Barbara, CA, USA, 2010, pp. 216-229.

[57] M. Caccamo, L.Y. Zhang, L. Sha and G. Buttazzo, “I-EDF: An implicit prioritized access protocol for wireless sensor networks,” *In Proc. 23rd IEEE International Real-Time Systems Symposium (RTSS02)*, 2002, pp. 39-48.

[58] T. Watteyne, I. Aug-Blum and S. Ubda, “Dual-mode real-time MAC protocol for wireless sensor networks: a validation/simulation Approach,” *In Proc. InterSense*, 2006.

[59] K.K. Chintalapudi, and L. Venkatraman, “On the Design of MAC Protocols for Low-Latency Hard Real-Time Discrete Control Applications over 802.15.4 Hardware,” *in Proc. 7th ACM/IEEE Int. Conf. Information Processing in Sensor Networks*, St. Louis, MO, USA, 2008, pp. 356-367.

[60] V. Namboodiri, and A. Keshavarzian, “Alert: An Adaptive Low-Latency Event-Driven MAC Protocol for Wireless Sensor Networks,” *in Proc. 7th ACM/IEEE Int. Conf. Information Processing in Sensor Networks*, St. Louis, MO, USA, 2008, pp. 159-170.

[61] M. Salajegheh, H. Soroush, and A. Kalis, “HyMAC: Hybrid TDMA/FDMA Medium Access Control Protocol for Wireless Sensor Networks,” *in Proc. 18th IEEE Int. Symp. Personal, Indoor and Mobile Radio Communications*, Athens, Greece, 2007, pp. 1-5.

[62] R. Biswas, V. Jain, C. Ghosh, and D. P. Agrawal, “On-Demand Reliable Medium Access in Sensor Networks,” *in Proc. 7th IEEE Int. Symp. a World of Wireless, Mobile and Multimedia Networks*, Buffalo, NY, USA, 2006, pp. 251-257.

[63] V. Jain, R. Biswas, and D.P. Agrawal, “Energy-Efficient and Reliable Medium Access in Sensor Networks,” *in Proc. 8th IEEE Int. Symp. a World of Wireless, Mobile and Multimedia Networks*, Helsinki, Finland, 2007, pp. 1-8.

[64] X. Wang, X. Wang, X. Fu, G. Xing and N. Jha, “Flow-Based Real-Time Communication in Multi-Channel Wireless Sensor Networks,” *the 6th European Conference on Wireless Sensor Networks (EWSN’09)*, Cork, Ireland, February 2009.

[65] X. Wang, X. Wang, G. Xing and Y. Yao, “Exploiting Overlapping Channels for Minimum Power Configuration in Real-Time Sensor Networks,” *7th European Conference on Wireless Sensor Networks (EWSN’10)*, Coimbra, Portugal, February 17-19, 2010, pp 97-113.

[66] G.-S. Ahn, S. G. Hong, E. Miluzzo, A. T. Campbell and F. Cuomo, “Funneling-MAC: A Localized, Sink-Oriented MAC For Boosting Fidelity in Sensor Networks,” *in 4th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Boulder, Colorado, USA, November 1-3, 2006, pp. 293306.

[67] I. Rhee, A. Warrier, M. Aia, J. Min, and M.L. Sichitiu, “Z-MAC: a hybrid MAC for wireless sensor networks,” *IEEE/ACM Trans. Netw.*, June 2008, Volume: 16 Issue:3, pp. 511-524.

[68] S. Mehta and, K. Sup Kwak, “H-MAC: A Hybrid MAC Protocol for Wireless Sensor Networks,” *International Journal of Computer Networks and Communications*, 2010, vol. 2, pp. 108-117.

[69] H. Pham and S. Jha, “An Adaptive Mobility-Aware MAC Protocol for Sensor Networks (MS-MAC),” *in IEEE MASS*. Fort Lauderdale, USA, October 2004, pp. 588-560.

[70] R. Kuntz, J. Montavont, and T. Nol, “Improving the medium access in highly mobile Wireless Sensor Networks,” *Journal of Telecommunication Systems*, August 2011, Vol. 47, No. 3-4.

[71] B.M. Khan and F.H. Ali, “Collision Free Mobility Adaptive (CFMA) MAC for wireless sensor networks,” *Journal of Telecommunication Systems*, August 2011, Vol. 47, No. 3-4.

[72] Online Survey on Energy-Efficient MAC Protocols, Available: *https://apstwo.st.ewi.tudelft.nl/ koen/MACsoup/*

[73] G. C. Buttazzo, “Hard Real-Time Systems: Predictable Scheduling Algorithms and Applications,” *Second Edition Springer Book*, University of Pavia, Italy, Ed. 2005.

[74] J. D. Decotignie, “Keynote: Real-time and wireless sensor networks: Why do we need another view at it ? ,” *in 7th International Workshop on RealTime Networks (RTN) (abstract)*, July 2008, pp. 67.

[75] R. S. Oliver, G. Fohler, “A Proposal for a Notion of Timeliness in Wireless Sensor Networks,” *Proc. 8th International Workshop on Real-Time Networks RTN’09*, Dublin, Ireland, June 2009.

[76] Y. Li, C. S. Chen, Y. Song,Z. Wang, “REAL-TIME QOS SUPPORT IN WIRELESS SENSOR NETWORKS: A SURVEY,” *7th IFAC Fieldbuses and Networks in Industrial and Embedded Systems*, Toulouse, France, November 7-9, 2007.

[77] R. Jurdak, P. Baldi, and C. Lopes, “Energy-Aware Adaptive Low Power Listening for Sensor Networks,” *in International Workshop on Networked Sensing Systems (INSS)*, San Diego, CA, USA, June 2005, pp. 2429.

[78] R. Braynard, A. Silberstein, and C. Ellis, “Extending Network Lifetime Using an Automatically Tuned Energy-Aware MAC Protocol,” *Proc. European Workshop on Wireless Sensor Networks (EWSN)*, Zurich, Switzerland, 2006, pp. 244-259.

[79] H. Wang, X. Zhang, F. Nat-Abdesselam, and A. A. Khokhar, “DPS-MAC: An Asynchronous MAC Protocol for Wireless Sensor Networks,” *In Proc. 14th International Conference of High Performance Computing (HiPC’07)*, Springer-Verlag Berlin, Heidelberg, 2007, pp. 393-404.

[80] I. Joe and H. Ryu, "A Patterned Preamble MAC Protocol for Wireless Sensor Networks," *IEEE International Conference on Computer Communications and Networks (ICCCN07)*, Honolulu, HI, 2007, pp. 1285-1290.

[81] P. Kumar, M. Gunes, Q.Mushtaq, and B.Blywis,"A real-time and energy-efficient MAC protocol for wireless sensor networks," *International Conference on Wireless and Optical Communications Networks (WOCN'09)*, Cairo , Egypt, April 28-30, 2009, pp. 1-5.

[82] S. Mahlknecht and M. Boeck, "CSMA-MPS: A Minimum Preamble Sampling MAC Protocol for Low Power Wireless Sensor Networks," *in IEEE Workshop on Factory Communication Systems*, Vienna, Austria, Sept. 22-24, 2004. pp. 73-80.

[83] E.-Y. Lin, J. Rabaey, and A. Wolisz, "Power-Efficient Rendez-vous Schemes for Dense Wireless Sensor Networks," *in IEEE International Conference on Communications*, Paris, France, June 20-24, 2004, Vol. 7, pp. 3769 3776.

[84] L. Bernardo, R. Oliveira, M. Pereira, M. Macedo, and P. Pinto, "A Wireless Sensor MAC Protocol for Bursty Data Traffic," *in 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Athens, Greece, Sept. 3-7, 2007, pp. 1-5.

[85] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing Sensor Networks in the Energy-Latency-Density Design Space," *IEEE Trans. Mobile Comput.*, January-March 2002, Vol. 1, No. 1, pp. 7080.

[86] D. Moss and P. Levis, "BoX-MAC: Exploiting Physical and Link Layer Boundaries in Low-Power Networking," *Technical Report SING-08-00*, 2008.

[87] K.-J. Wong and D. Arvind, "Speckmac: Low-power decentralized MAC protocol low data rate transmissions in Specknets," *Proc. 2nd International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality (REALMAN'06)*, (ACM SIGMOBILE), Italy, May 2006, pp. 71-78.

[88] C.J. Merlin and W.B. Heinzelman, "Schedule Adaptation of Low-Power-Listening Protocols for Wireless Sensor Networks," *IEEE Trans. Mobile Computing*, May 2010, Vol. 9, Issue: 5, pp 672685.

[89] D. Lee and K. Chung, "RA-MAC: An Energy Efficient and Low Latency MAC Protocol Using RTS Aggregation for Wireless Sensor Networks," *International Conference on Advanced Technologies for Communications (ATC)*, Hanoi,Oct. 6-9, 2008, pp 150-153.

[90] S. IGaneriwal, I. Tsigkogiannis, H. Shim, V. Tsiatsis, M.B. Srivastava, D. Ganesan, "Estimating Clock Uncertainty for Efficient Duty-Cycling in Sensor Networks," *IEEE/ACM Trans. Netw.*, June 2009, Vol. 17, No. 3, pp. 843-856.

[91] P. Hurni, T. Braun, and M. Anwander, "Evaluation of WiseMAC and Extensions on Wireless Sensor Nodes," *Springer Telecommunication Systems Journal*, Springer US, September4, 2009, Vol.43, No.pp. 1-2, ISSN1018-4864.

[92] P. Hurni, and T. Braun, "MaxMAC: A Maximally Traffic-Adaptive MAC Protocol for Wireless Sensor Networks," *European Conference on Wireless Sensor Networks (EWSN)*, Coimbra, Portugal, February 17-19, 2010, pp 289-305.

[93] J. Kim, J. On, S. Kim, and J. Lee, "Performance Evaluation of Synchronous and Asynchronous MAC Protocols for Wireless Sensor Networks," *Second International Conference on Sensor Technologies and Applications (SENSORCOMM'08)*, Cap Esterel, France, Aug. 25-31, 2008, pp. 500-506.

[94] Y.-C. Chang, J.-R. Jiang, J.-P. Sheu, and H.-Y. Shih, "ADCA: An Asynchronous Duty Cycle Adjustment MAC Protocol for Wireless Sensor Networks," *Global Telecommunications Conference IEEE GLOBECOM 08*, New Orleans, LO, Nov.30-Dec. 4, 2008, pp. 383-3870.

[95] J. Polastre, "A Unifying Link Abstraction for Wireless Sensor Networks," *Ph.D. dissertation*, University of California, Berkeley, October 2005.

[96] X. Shi and G. Stromberg, "SyncWUF: An Ultra Low-Power MAC Protocol for Wireless Sensor Networks," *IEEE Trans. Mobile Computing*, Jan. 2007, Vol. 6, Issue 1, pp. 115-125.

[97] M. Miller and N. Vaidya, "A MAC Protocol to Reduce Sensor Network Energy Consumption Using a Wakeup Radio," *Trans. Mobile Computing*, May-June 2005, Vol. 4, No. 3, pp. 228242.

[98] Q. Yu, C. Tan, H. Zhou, "A Low-Latency MAC Protocol for Wireless Sensor Networks," *International Conference on Wireless Communications, Networking and Mobile Computing (WiCom07)*, Shanghai, Sept. 21-25, 2007, pp. 28162820.

[99] I. Gragopoulos, I.Tsetsinas, E.Karapistoli, and F.-N.Paulidou, "Performance Study of a New MAC Algorithm to Reduce Energy and Delay in Wireless Sensor Networks," *New Technologies, Mobility and Security*, Springer, 2007, pp. 3951.

[100] B. lang, l.B. Lim, M.L. Sichitiu,"AS-MAC: An asynchronous scheduled MAC protocol for wireless sensor networks," *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MAHSS)*, Atlanta, GA, Oct. 2, 2008, pp. 434441.

[101] W.-H. Liao and H.-H. Wang, "An asynchronous MAC protocol for wireless sensor networks," *Journal of Network and Computer Applications*, 2008, Vol. 31, No. 4, pp. 807-820.

[102] T. Zheng, S. Radhakrishnan, and V. Sarangan, "PMAC: an adaptive energy-efficient MAC protocol for wireless sensor networks," *Proc. 19th IEEE International Parallel and Distributed Processing Symposium (IDPDS)*, April 4-8, 2005, pp. 8.

[103] K. Jamieson, H. Balakrishnan, and Y. C. Tay, "Sift: A MAC Protocol for Event-Driven Wireless Sensor Networks," *Third European Conference on Wireless Sensor Networks (EWSN06)*, Zurich, Switzerland, February 2006, Vol. 3868, pp. 260-275.

[104] I. Demirkol and C. Ersoy, "Energy and delay optimized contention for wireless sensor networks," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, August 2009, Vol. 53,Issue 12, pp. 2106-2119.

[105] Y. Sun, O. Gurewitz, and D.B. Johnson, "RI-MAC: a receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks," *Proc. 6th ACM conference on Embedded network sensor systems (SenSys'08)*, 2008, pp. 1-14.

[106] P. Dutta, S. Dawson-Haggerty, Y. Chen, C. Liang, and A. Terzis, "Design and Evaluation of a Versatile and Efficient Receiver-Initiated Link Layer for Low-Power Wireless," *In Proc. 8th ACM Conference on Embedded Networked Sensor Systems (Sensys'10)*, Zurich, Switzerland, Nov. 3-5, 2010.

[107] P.G. Park, C. Fischione, A. Bonivento, K.H. Johansson, and A. Sangiovanni-Vincentelli, "Breath: a self-Adapting protocol for wireless sensor networks in control and automation," *IEEE SECON*, San Francisco, CA, USA, 2008, pp. 323-331.

[108] S. Rashwand, J.V. Misic, V.B. Misic, S. Biswas, and Md.M. Haque, "A Novel Asynchronous, Energy Efficient, Low Transmission Delay MAC Protocol for Wireless Sensor Networks, " *ICDCS Workshops Proceedings*, 2009, pp. 186-193.

[109] S. Liu, K.-W. Fan, and P. Sinha, "CMAC: An Energy Efficient MAC Layer Protocol Using Convergent Packet Forwarding for Wireless Sensor Networks," *in SECON*, San Diego, CA, USA IEEE, June 2007.

[110] M. Zorzi and R. Rao, "Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Energy and Latency Performance," *IEEE Trans. Mobile Computing*, 2003, Vol. 2, No. 4, pp. 349365.

[111] J. Kim, X. Lin, N.B. Shroff, and P. Sinha, "On Maximizing the Lifetime of Delay-Sensitive Wireless Sensor Networks with Anycast," *in Proc. IEEE INFOCOM*, Pheonix, AZ, April 2008, pp. 807-815.

[112] J. Kim, X. Lin, and N.B. Shroff, "Optimal Anycast Technique for Delay-Sensitive Energy-Constrained Asynchronous Sensor Networks," *INFOCOM*, 2009, pp. 612-620.

[113] I. Aad and C. Castelluccia, "Differentiation Mechanisms for IEEE 802.11," *In Proc. IEEE INFOCOM' 01 (S0743-166X)*, April 2001, pp. 209-218.

[114] G. EkbataniFard, and R. Monsefi, "A Detailed Review of Multi-Channel Medium Access Control Protocols for Wireless Sensor Networks," *International Journal of Wireless Information Networks*, Jul. 2011, Vol. 19, pp. 1111-1132.

Messaoud Doudou obtained his Engineer Degree in computing science from National Institute of Computer science (INI-Algiers) in 2005. After two years, he obtained his master degree from the University of Science and Technology USTHB Algiers in 2008, where he worked mainlly on Wireless networks and especially on wireless mesh networks. Currently, he is a PHD student at the USTHB University. He is also a permanent Full-time Researcher at the Center of Research on Scientific and Technical Information (CERIST) in Algiers. He is a research member of SensLab research group at CERIST research center. His research interest includes medium access control, wireless sensor networks, and QoS in wireless networks.

Djamel Djenouri obtained the Engineer Degree, the Master Degree, and the PhD in Computer Science from the University of Science and Technology USTHB, Algiers, Algeria, respectively in 2001, 2003 and 2007. During his PhD he visited John Moores University in Liverpool, UK, where he carried out collaborative work with researchers of the “Distributed Multimedia Systems and security” group. From 2008 to 2009, he was granted a Post-doctoral Fellowship from the European Research Consortium on Informatics and Mathematics (ERCIM), and he worked at the Norwegian University of Science and Technology (NTNU), in Trondheim, Norway, where he participated in the MELODY project supported by the Norwegian Research Council. Currently, he is a permanent Full-time Researcher at the CERIST Research Center in Algiers. His researches focus on ad hoc and sensor networking, especially on the following topics: security, power management, routing protocols, MAC protocols, and vehicular ad hoc networks. He participated in many international conferences and published more than 30 papers in international journals and proceedings. He is a member of the ACM and served as TPC member of many international conferences. He also served as reviewer for many international journals, and chaired workshops organized with IEEE DCOSS 2010 and GlobCom 2010. In 2008, he was granted the best publication award from ANDRU, supported by the Algerian government.

Nadjib Badache was with CISTTT (Centre dInformation Scientifique et Technique et de Transfert Technologique) now CERIST (Centre de Recherche sur l’Information Scientifique et Technique) at Algiers, from 1978 to 1983, working on applied research projects in information retrieval, operating systems and library management. In 1983, he joined USTHB University of Algiers, as Assistant Professor and then Professor, where he teaches operating systems design, distributed systems and networking with research mainly in distributed algorithms and mobile systems. From 2000 to 2008, he was head of LSI (Laboratoire des Systmes Informatiques of USTHB University) where he conducted many projects on routing protocols, energy efficiency and security in Mobile Adhoc Networks and Wireless Sensor Networks. Since March 2008, he is CEO of CERIST and Professor at USTHB University.