CL-MAC: Cross-layer MAC Protocol for Delay Sensitive Wireless Sensor Network Applications

Kechar Bouabdellah and Sekhri Larbi
University of Oran
Algeria

1. Introduction

Recent advances in micro-electromechanical systems (MEMSs) technology, wireless communications field and nanotechnology have enabled the design of low-power, low cost smart sensor nodes equipped with multiple onboard functions such as sensing, computing, and communications. Such intelligent devices networked through wireless links have been referred to as Wireless Sensor Networks (WSN). The basic function of the network is to observe some phenomenon by using the sensors and communicate the sensed data to a common destination called the base station or the Sink. In most application scenarios, sensor nodes are powered by small batteries, which are practically non-rechargeable, either due to cost limitations or because they are deployed in hostile environments.

Many WSN applications, that are delay sensitive in case when an abnormal event occurs, exist in practice: environmental monitoring (for example forest fire detection, intruder detection), assistance for old or disabled people and structural health monitoring. In these applications, the detected event is considered as an urgent data which must be transmitted quickly towards the Sink for fast intervention. To achieve this requirement, it is necessary to decrease latency at MAC layer when transmitting urgent data from the source node to the Sink.

These considerations motivate well energy saving and low latency WSN designs.

Many research works have been developed for energy efficiency at each layer of the protocol stack by proposing new algorithms and protocols. In particular, the MAC layer was of great interest for many researchers because it was considered as an important source of energy waste. It is summarized in (Zhi-Wen et al., 2005; Injong et al., 2005; Muneeb et al., 2006; Sohraby et al., 2007):

- *Overhearing*: a sensor node receives packets that are transmitted for other nodes. This is mainly due to the radio transmission nature (omni-directional) forcing every node of the neighborhood to waste energy when receiving and decoding these packets. These packets are eventually dropped after the node realizes that the destination address is different from its own address.
- *Collision*: since the radio channel is shared by many nodes, a collision takes place every time when two nodes try to send their packets at the same time. Collisions increase energy consumption and latency in case of packets deliverance mechanism due to retransmissions.

- *Control packets (overhead)*: packet headers and control packets (*RTS/CTS/ACK*) used by a MAC protocol do not contain application data, thus they are considered as supplementary data (*overhead*). Control packets can be of importance since most applications use data packets with reduced size.
- *Idle listening*: it is a dominant factor for energy waste in WSN. Indeed, when a node is not in the transmission mode, it must continuously listen to the channel in order to receive possible traffic that is not sent. In this case, the amount of energy waste is almost equal to the energy dissipated by a normal reception according to (Wei et al., 2004) (the ratios of E_{idle}:$E_{receiving}$:$E_{transmitting}$ are 1:1.05:1.4).
- *Over emitting*: this case occurs when a sensor node receives a packet while it is not ready. This situation forces the sender to perform new retransmissions that are strongly linked to synchronisation problem and therefore wastes energy.

In order to decrease or at least eliminate these various sources of energy waste, several protocols have been proposed these last years. They can be divided into two main classes: TDMA-based MAC protocols and Contention-based MAC protocols.

1.1 TDMA-based MAC protocols

These protocols (known as deterministic) are employed to avoid collisions by exclusively allocating time slots to sensor nodes. However, these protocols require the presence of a management authority (for example a dedicated access point) to regulate the access to the medium by broadcasting a schedule that specifies when, and for how long, each controlled sensor node may transmit over the shared channel. In these protocols, the channel is divided into time slots, which are grouped into logical frames (see Fig. 1 in which a set of N contiguous slots form a logical frame). In each logical frame each sensor node is assigned a set of specific time slots. This set constitutes the schedule according to which the sensor node operates in each logical frame.

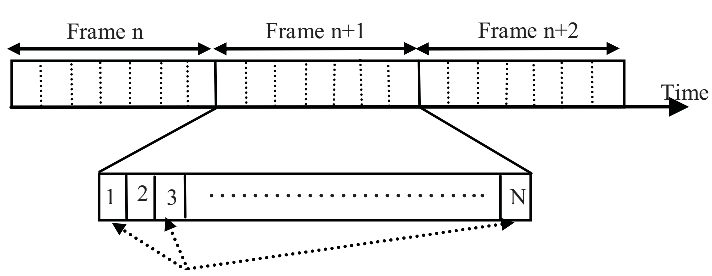

Fig. 1. Logical frame in TDMA-based protocols for WSN

The schedule can be either fixed, constructed on demand on a per-frame basis by the base station or hybrid (Sohraby et al., 2007). Outside these assigned slots, a sensor node goes to sleep mode in which the radio transceiver is completely turned off to conserve energy. However, in WSN we need distributed protocols to allocate time slots to sensor nodes (Willig, 2006), but such distributed schemes tend to be somewhat complex (see for example SMACS (Sohrabi & Pottie, 1999; Sohrabi et al., 2000), TRAMA (Rajendran et al., 2003) or LEACH (Heinzelman et al., 2002)). Network topology changes (due for example to sensor nodes running out of energy, the deployment of new nodes or node mobility) require the slot allocation protocol been executed periodically. In addition, TDMA-based MAC schemes

require tight time synchronization between nodes to avoid overlap of time slots. This in turn requires continuous execution of a time synchronization protocol.
This makes the use of these protocols more complex in WSN where each node has, in general, no priority assigned and very limited resources.

1.2 Contention-based MAC protocols

These protocols known as CSMA-based are usually used in multi-hop wireless networking due to their simplicity and their adequacy to be implemented in a decentralized environment like WSN. When these protocols are used, collisions can be occur in case of a receiver is located in the radio range of at least two sensor nodes transmitting simultaneously data packets to it. Collisions waste the energy of both the transmitter and the receiver and as a result packet retransmissions can occur which create additional load for a congested channel. In CSMA-based protocols, collisions are often the result of hidden terminal problem. Consider the situation in Fig. 2 where A and B can hear each other, B and C can hear each other but A and C cannot. Nodes A and C both want to transmit a packet to their common neighbour B. Both nodes sense an idle channel and start to transmit their packets. The signals of nodes A and C overlap at B and are destroyed (collision problem).

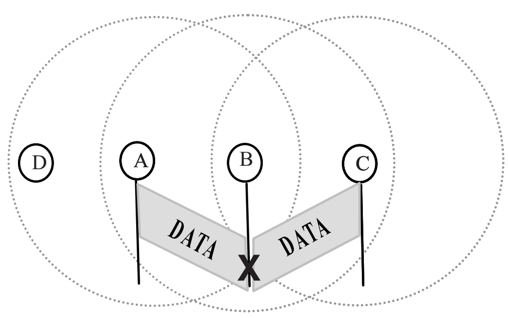

Fig. 2. Hidden terminal scenario

To reduce these collisions in ad-hoc (sensor) networks, the 802.11 standard defines a virtual carrier mechanism based on the Request-To-Send (RTS)/(Clear-To-Send) CTS scheme defined in MACA protocol (MultiAccess Collision Advoidance) (Karn, 1990). By using this scheme, collisions between hidden nodes at common neighbors can be avoided. A sensor node (node A in Fig. 3) wanting to transmit a unicast packet initiates a handshake by transmitting an RTS control packet after a specified time called the Distributed Inter Frame Space (DIFS). The receiver (node B in Fig. 3)) waits a Short Inter Frame Space (SIFS) before responding by sending a CTS control packet, which informs all its neighbors of the upcoming transfer. Since the SIFS interval is set shorter than the DIFS interval, the receiver takes precedence over any other sensor node attempting to send a packet (Koen & Gertjan, 2004). The effective DATA transfer (from A to B) is now guaranteed to be collision free. So, after a SIFS period, DATA packet is transmitted by sender (A) and receiver (B) waits a SIFS period before acknowledging the reception of the data by sending an ACKnowledgement control packet (ACK). If sender (A) does not receive the ACK packet, it assumes that the data was lost due to a collision at receiver (B) and enters a binary exponential backoff procedure. This same procedure can be used when two RTS packets collide, which is technically still possible. The RTS/CTS control packets specify in their header the duration

of the upcoming DATA/ACK sequence which allow neighboring nodes to overhear the control packets, set their Network Allocation Vector (NAV) and defer transmissions until it expires (NAV(RTS) at neighbor node D and NAV(CTS) at neighbor node C in Fig. 3). The radio can be switched off for the duration NAV to save energy waste. Note that, all broadcast and multicast packets are transmitted without RTS/CTS scheme.
In the contention-based MAC protocols, the RTS/CTS scheme is sufficient to greatly reduce collisions and increase bandwidth utilization, but unfortunately does not completely solve the hidden terminal problem, according to (Sohraby et al., 2007).

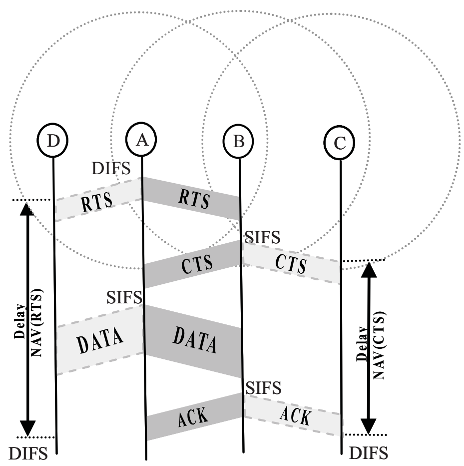

Fig. 3. Collision avoidance using RTS/CTS handshake (Sohraby et al., 2007)

A number of MAC protocols have been developed to considerably reduce other sources of energy waste. They focus on reducing idle listening, but collisions, control packets (overhead) and overhearing are also addressed.
The first approach to reduce idle listening is based on preamble sampling and operates at the physical layer (see for example (Halkes et al., 2005; Bachir et al., 2006; EL-Hoiydi, 2002; Polastre et al., 2004; Buettner et al., 2006). In this asynchronous approach, as illustrated in Fig. 4, a receiver turns on periodically the radio to sample for incoming data and detects if a preamble is present or not (Halkes at al., 2005). If it detects a preamble, it will continue listening until the start-symbol arrives and the message can be properly received. If no preamble is detected the radio is turned-off again until the next sample.

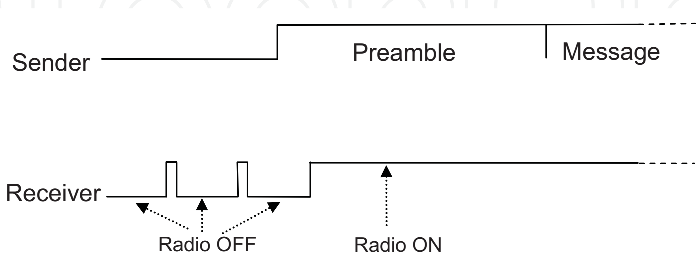

Fig. 4. A sender uses a longer preamble to allow the receiver to only turn its radio on periodically (Halkes et al., 2005)

The second approach trying to mitigate the idle listening uses wake-up/sleep mechanisms and/or *RTS/CTS/DATA/ACK* signalling scheme from 802.11x standard to reduce collision, overhearing and control packet (overhead). A well know MAC protocols in the literature using this approach are S-MAC, T-MAC (with the automatic adaptation of the duty cycle to the network traffic), D-MAC and Z-MAC (see section 2.1 for more detail).
Recently, a new generation of MAC protocols (*Cross-layer MAC protocols)* using several layers in order to optimize energy consumption has emerged. These layers can be exploited into two modes: *interaction* or *unification* as depicted in Fig. 5. In the interaction mode, the MAC protocol is built by exploiting the data generated by other adjacent layers. MAC-CROSS Protocol (Suh et al., 2006) is an example of Cross-layer approach which allow the routing information of the network layer to be exploited by the MAC layer (interaction between MAC and network layers) by leaving only the communicating nodes in activity and by putting into sleep mode the other neighbor nodes (not concerned by this communication). In order to avoid collisions, MAC-CROSS uses the control messages *RTS/CTS/ACK*. On the other hand, a Cross-layer design mode by unification requires the development of only one layer including at the same time the functionalities of considered layers.

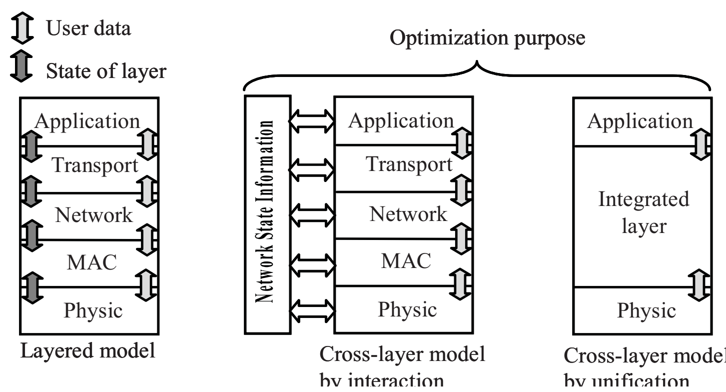

Fig. 5. Cross-layer model illustration

In this chapter, we propose a Cross-layer protocol named CL-MAC, based on the same ideas used by MAC-CROSS. The fundamental difference between our proposal and MAC-CROSS lies on the level of the number of consecutive nodes that are implied in MAC functioning at each frame. Indeed, MAC-CROSS acts on three consecutive communicating nodes while CL-MAC uses all the nodes included in a given routing path from the source node to the Sink in one frame. Two main operations take place simultaneously in this routing path after an RTS/CTS exchange at the beginning of each data transmission: on one hand successive transmissions of CTS packets which advance quickly towards the Sink in order to reserve a path. All nodes included in this path remain in activity and all other nodes in their vicinity enter sleep mode for a given time interval. On the other hand DATA/ACK packet exchanges between communicating nodes in the routing path (a relatively slow process) which advance progressively.
Temporal Petri nets are introduced in order to model underlying operation of the proposed protocol and the TiNA tool is carried out for analytical validation of some related properties. A comparative study between CL-MAC, MAC-CROSS and S-MAC in term of energy saving and low latency has been performed for evaluation purpose by using a home simulator.

The rest of the chapter is organized as follow: in the next section, we introduce main works in literature related to energy saving at the MAC layer level. Some OSI-based protocols and others based on Cross-layering approach are given in this section. In section 3, we give more details about the proposed protocol CL-MAC. In Section 4, we present a formal representation of the CL-MAC protocol using time Petri nets modeling approach. The analytical validation of some properties of CL-MAC using the TiNA software tool is given in section 5. The performance evaluation of CL-MAC protocol by comparison with a similar MAC protocols like S-MAC and MAC-CROSS is presented in section 6. Finally, we conclude our work and discuss some future perspectives.

2. Related work

In this section, we present some MAC layer protocols developed some years ago that enable energy conservation in WSN. First, compatible OSI protocols are presented and followed by two important cross-layer protocols: MAC-CROSS and XLM. Especially, MAC-CROSS protocol is considered as a basis of the development of our proposal.

2.1 Compatible OSI protocols

Many studies in WSN have showed that energy consumption during a communication is four times greater than the energy consumed in both the processing and sensing operations. This fact lead communication protocols designers to take a particular interest into the WSN-MAC layer and to propose some original ideas to efficiently manage that layer. The medium access must take into account all sources of energy waste considerations.

Sensor-MAC (S-MAC) protocol is a very popular protocol developed at California University (Wei et al., 2002; Koen & Gertjan, 2004; Zhi-Wen et al., 2005). Its main objective is to conserve energy in WSN and it takes into consideration that fairness and latency are less critical issues compared to energy conservation. The basic idea behind S-MAC is the management of local synchronizations and the schedule of sleep/listen periods based on these synchronizations. Neighboring nodes form virtual clusters in which they periodically broadcast special SYNC messages to keep synchronized. The period for each node to send a SYNC packet is called the synchronization period. If two neighboring nodes reside in two virtual clusters, they wake up at the listen periods of both clusters (Demirkol, 2006). Every frame in S-MAC as shown in Fig. 6 is divided into an active period and a sleep period. The active period is divided into three parts for SYNC, RTS and CTS packets. In this figure, nodes 1 and 3 are synchronized to the schedule of node 2 by receiving its SYNC packet. This means that nodes 1, 2 ad 3 share a same virtual cluster. Node 3 initiates an RTS/CTS exchange with node 1 to transmit data. When CTS packet is received, data transmission will immediately follow. Nodes 1 and 3 stay active until the completion of data transfer, whereas node 2 follows its normal sleep schedule. In the sleep period, when data transmission ends, communicating nodes enter sleep mode by switching off their radio transceivers. Collision avoidance is achieved by a carrier sense (CS in Fig. 6).

S-MAC also includes message passing support in which long messages are divided into frames and sent in a burst-mode. In this case, only one RTS and one CTS are used to reserve the medium for the time needed to transmit all fragments.

Several other energy efficient protocols in the literature are based on wake-up/sleep mechanism: T-MAC (Koen & Gertjan, 2004), D-MAC (Lu et al., 2004), Z-MAC (Injong et al.,

2005). T-MAC and D-MAC are considered similar to the S-MAC protocol with an adaptive duty cycle.
The T-MAC (Timeout-MAC) protocol adapts the duty cycle to the network traffic in order to improve S-MAC, but instead of using a fixed length active period, T-MAC uses a time-out

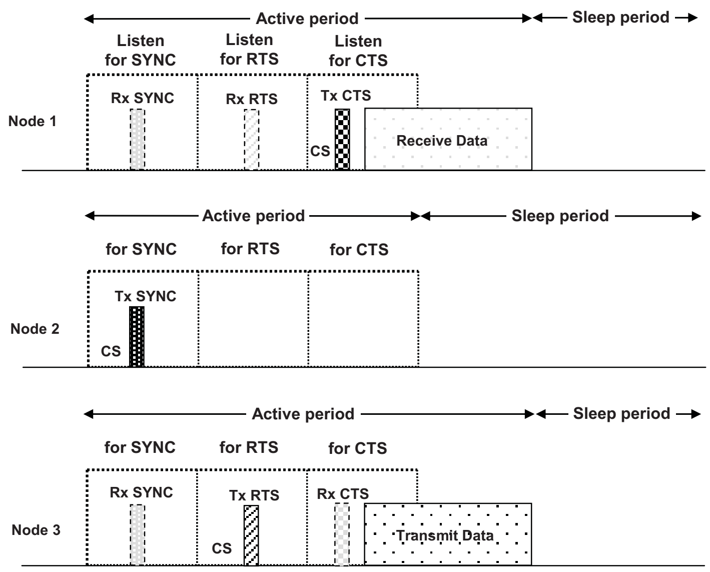

Fig. 6. Timing relationship between different sensor nodes in SMAC (This figure was redrawn from (Dewasurenda & Mishra, 2005))

mechanism to dynamically determine the end of the active period (Halkes, 2005). In Fig. 7, when a node does not detect any activity within the activity time-out period (TA), it can safely assume that no neighbor wants to communicate with it and then enters sleep period. If the node engages or overhears a communication, it simply starts a new TA period after that communication finishes.
The D-MAC (Data-gathering MAC) protocol includes an adaptive duty cycle like T-MAC for energy efficiency and ease of use. In addition, it provides low node-to-Sink latency, which is achieved by supporting convergecast communication paradigm that is the mostly observed communication pattern within sensor networks. DMAC achieves very good latency compared to other sleep/listen period assignment methods, but unfortunately collision avoidance methods are not utilized.

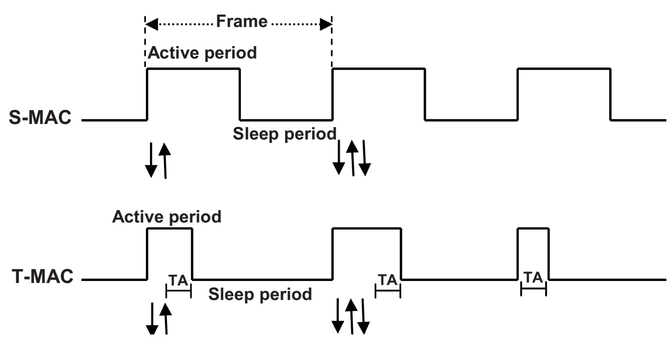

Fig. 7. The S-MAC and T-MAC duty cycles; the arrows indicate transmitted and received messages; note that messages come closer together (TA: *Activity Time-out period*) (Halkes et al., 2005)

Z-MAC (Zebra MAC) is a hybrid MAC protocol for wireless sensor networks that combines the strengths of TDMA and CSMA while offsetting their weaknesses. The main feature of Z-MAC is its adaptability to the level of contention in the network so that under low contention, it behaves like CSMA, and under high contention, like TDMA. A distinctive feature of Z-MAC is that its performance is robust to synchronization errors, slot assignment failures and time-varying channel conditions. Z-MAC is used as the default MAC for Mica2 mote.

2.2 Cross-layer protocols

Other protocols based on OSI layer models try to reduce problems encountered in WSN. Network layer protocols tend to optimize paths between network's nodes and the Sink while application layer protocols try to obtain correct, accurate and compressed and/or aggregated information (Holger et al., 2003; Cheng et al., 2006) so as to reduce the amount of packets in the network. OSI-based protocols are not flexible, not optimal and consequently reduce network performances. To mitigate these drawbacks, a new MAC approach based on interaction or unification of two or more adjacent layers, called Cross-layer MAC optimization, has emerged (Akyildiz & Ismail, 2004; Suh et al., 2006; Akyildiz et al., 2006). Some protocols using a Cross-layer technique in medium access control layer can be found in literature such as MAC-CROSS (Suh et al., 2006) and XLM (Akyildiz et al., 2006).

- *MAC-CROSS protocol:* in this protocol, only a few nodes concerned with the actual data transmission are asked to wake-up, while other nodes that are not included on a routing path and hence are not involved in the actual transmission at all. In exchanging *RTS* and *CTS* packets, a field corresponding to a final destination address is added. The neighborhood nodes belonging to the path extend their wake-up time while other nodes prolong their sleep time.
- *XLM* (*Cross-Layer Module for Wireless Sensor Networks*) *protocol:* this protocol proceeds differently comparing to others traditional architecture based protocols for WSN. The communication in XLM is based on the *initiative* concept considered as the core of XLM and implicitly incorporates the intrinsic functionalities required for successful communication in WSN. A node starts a transmission by transmitting to his neighborhood an *RTS* packet to indicate that it has a packet to send. Upon receiving an *RTS* packet, each neighborhood node decides to participate to the communication by determining an *initiative I* defined as follows:

$$I = \begin{cases} 1, \ if \begin{cases} \xi_{RTS} \geq \xi_{Th} \\ \lambda_{relay} \leq \lambda_{relay}^{Th} \\ \beta \leq \beta^{\max} \\ E_{rem} \geq E_{rem}^{\min} \end{cases} \\ 0, \quad otherwise \end{cases} \tag{1}$$

ξ_{RTS}: Signal Noise Ratio (*SNR*) received from *RTS* packet,
λ_{relay}: packets rate transmitted by relay and by node,
β: node's buffer occupation,
E_{rem}: node's residual energy.
Values in the right side of inequalities give us respective thresholds and initiative *I* is initialised to 1 if all conditions illustrated in (1) are satisfied.
1. The first condition ensures that reliable links are built for the communication.
2. The second and third conditions are employed by local control congestion. Second condition prevents congestion by limiting the transmitted traffic by a relay node, while the third condition ensures that no buffer overflow exists for this node.
3. The last condition ensures that the residual energy E_{rem} of a node do not exceed a minimal threshold $E_{rem}^{\min}$.
Cross-layering functionalities of the XLM protocol are represented by the constraints defined in the initiative I of a node enabling it to carry out a local congestion control, hop by hop reliability and distributed operation.
The component of local congestion control of XLM ensures energy efficiency and a reliable communication. Results of performance evaluation revealed that XLM is better than one-layer protocols in terms of communication processing and implementation complexity considerations.

3. CL-MAC protocol presentation

CL-MAC (*Cross-Layer MAC*) can be added to MAC protocols class by exploiting interactions between adjacent layers in order to minimize all energy waste sources and decrease latency during a multi-hop routing of a delay sensitive traffic from a particular source node towards the Sink in a WSN. The MAC layer enables access to the medium with wake-up/sleep schedule. Before presenting CL-MAC, we consider the following suppositions:
- Typical utilization scenario: to make clear our contribution, Fig. 8 shows a typical example of WSN architecture in which CL-MAC must be used. When a delay sensitive event has been detected (for example the detection event of forest fire around the area covered by sensor node S1 in Fig. 8), the data related to this event can be considered as an urgent traffic and must be delivered quickly to the Sink. CL-MAC acts at the MAC layer and aims to reserve a low latency and energy efficient path to deliver this urgent traffic from the source node generating this traffic towards the Sink. We can distinguish two types of nodes included in this path, in addition to the Sink: 1) a sensor node that collects sensed data which must be sensitive to the end-to-end delay (node S1 in Fig. 8), 2) relaying nodes (like

A or B in Fig. 8) that are responsible only for relaying a delay sensitive traffic. In the scenario showed by Fig. 9, the path S1-A-B-C-D-E-F-Sink is an example of an urgent

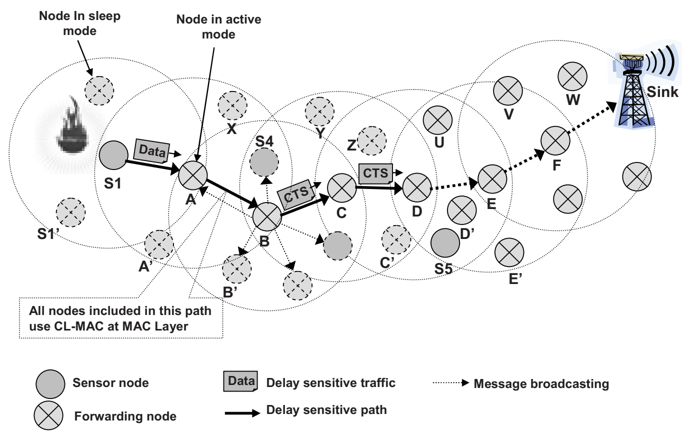

Fig. 8. Typical utilization scenario of CL-MAC

reserved path using CL-MAC. Each node included in this path like node B and its neighbors like nodes B' and S4 use CL-MAC at their MAC layer.

- The network is randomly deployed in a coverage area followed by a synchronisation phase.
- Nodes are locally and periodically synchronised like in Z-MAC protocol.

In the rest of the chapter, we refer to a delay sensitive traffic as *'urgent traffic'* or simply '*Data packet*' and to a delay sensitive path as '*urgent path*'.

In the following sections, we successively present the CL-MAC layer in the context of a cross layering approach, the new data structures of RTS and CTS control packets and their interpretation, the detailed algorithm for CL-MAC and its principal advantages and, finally, some other related details.

3.1 CL-MAC Layer

The neighborhood list established by each node contains information about neighboring nodes (identifier, position, and schedule table) and the routing table maintained by a routing agent in the network layer. The routing protocol used is based on the greedy approach, referred to as position-based routing, in which packet forwarding decision is achieved by utilizing location information about candidate nodes in the vicinity and the location of the final destination only. The distance-based Greedy forwarding scheme, proposed by Finn (Finn, 1987), has been adopted in our case. In this forwarding approach, a next hop node is the nearest neighbor node to the final destination (Sink). Only the nodes closer to destination than the current node are considered.

As illustrated in Fig. 9, CL-MAC operates at the MAC layer and the node implementing it can transmit two kind of unicast traffic: delay sensitive traffic generated locally (node implementing CL-MAC acts as a source node) or delay sensitive traffic received externally from a neighborhood (node implementing CL-MAC acts as a relaying node). When a source node acts as a relaying node and in the same time wants to transmit its local delay sensitive traffic, both kind of traffic can be transmitted towards the Sink.

CL-MAC exploits the routing information through a cross-layer design approach based on interaction strategy. In this strategy, CL-MAC interacts with the adjacent routing layer by using a simple Get/Store mechanism on a common memory storage dedicated to store the

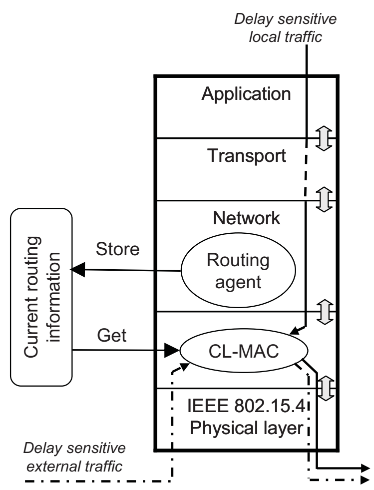

Fig. 9. CL-MAC : interaction between MAC and network layers

current routing information (see Fig. 9). This mechanism allows the routing agent to write in common memory information about the next hop forwarding node to which the current sender must forward a packet. This routing information can be obtained and exploited thereafter by CL-MAC to carefully perform a cross layering MAC operation. By knowing the next hop each time it advances in the routing path, CL-MAC can reserve a shared wireless medium during a time involved in each data communication.
In Fig. 9, we have proposed the IEEE 802.15.4 (IEEE, 2003) standard at the physical layer, which is dedicated at the origin for Low Rate-WPAN networks, but it is also very adapted for the WSN because it is designed for low-data-rate, low-power-consumption and low-cost applications.

3.2 New Data structure for RTS and CTS

Before describing further details about CL-MAC operations, one should be aware that weak modifications are made in *RTS* and *CTS* message structures without violating the IEEE 802.11 standard. The modified message structures of *RTS* and *CTS* illustrated by Fig. 10-(a) and (b) are proposed. The newly added field in RTS is the 'Next_Node_Adress', obtained by the sender's routing agent, and designates the address of the next node in the routing path to which packets must be transmitted. The new fields of CTS are 'Next_Node_Adress' and 'Sender_Adress'. The 'Next_Node_Adress' field has the same significance than 'Next_Node_Adress' of an RTS packet, but it is obtained by the routing agent of the receiver of the CTS packet. By specifying its address in a 'Sender_Adress' field of a sending CTS message, a node ignores receiving a CTS in which its address is specified as a previous node address. This change from the sender address to previous address has made in the receiver of the CTS message.
To make CL-MAC do that in a correct way, the Sink node address is supposed to be aware at the level of each node of the network in case of a mono-Sink WSN.

RTS	Frame control	Duration (NAV)	Previous node address	Next node address	Sender address	CRC

(a) The new RTS frame format in CL-MAC

CTS	Frame control	Duration (NAV)	Previous node address	Next node address	Sender address	CRC

(b) The new CTS frame format in CL-MAC

Fig. 10. Structure of *RTS* and *CTS* messages in CL-MAC (added fields)

However, it is important to note that *CTS* and *RTS* messages have identical structure but the difference is in their interpretation by the receiver node. Thus, if the receiver is:

- *next node*: the message acts as an *RTS* sent by the sender.
- *previous node*: the message acts as a *CTS* coming from a transmitter as a response to an *RTS*.
- *any other node*: the message controls the behaviour of the node and forces it to switch to sleep mode.

In a wake-up period, a node continues to listen to the medium for a short period. Two cases can be considered if the medium is not allocated:

Case 1. The node has urgent data to transmit towards the Sink, then takes immediately possession of the medium and informs its neighboring nodes of that decision.

Case 2. Node has no data to transmit. In this situation, it turns off its radio in order to avoid energy dissipation in idle listening and overhearing situations. In this way, the node takes more time in sleep mode than in S-MAC protocol (½ frame + communication time) and stays in this mode until the next frame corresponding to the current schedule.

If the medium is allocated in this period, this means that a communication is occurring or that another node is trying to get medium control. In this case two situations can occur again:

- *Situation 1*. Another node tries to transmit packets. Then, the Backoff algorithm (Ignatius, 2006) is used to resolve this contention problem and makes possible for only one node (elected node) to obtain the control access rights. All other nodes enter sleep mode except the receiver of the packet which remains awake in order to communicate with the elected node.
- *Situation 2*. The node has no data to send. Then, if it is concerned with the current communication it remains in wake-up mode, otherwise it enters into sleep mode until the next frame.

A receiver node is identified by the sender which refers to its routing table that contains all the information related to the path between the sender and the Sink (distance, hops number of each alternative path, previous node identifier).

3.3 Algorithm for CL-MAC

Fig. 11 provides a detailed algorithm in order to implement the CL-MAC protocol at the level of each node. In case of urgent traffic our protocol forces all neighboring nodes, which are not selected for routing paths, to switch to sleep mode.

Fig. 12 illustrates the temporal behaviour of CL-MAC corresponding to the routing path (S1-A-B-C-D-E-F-Sink) depicted by Fig. 8. In Fig. 12, the data transferring follows a unidirectional path from a source node (S1) to the Sink. This node (S1) begins by transmitting an RTS control packet to its next hop (A). When this control packet has been successfully received, node (A) replies with a CTS control packet back to sender (S1). All neighbors of the sender and the receiver turn off their radio to save energy (nodes S1' and A in Fig. 8) except the next hop of the receiver that stays active (node B in Fig. 8). When receiver (A) is ready to receive the DATA packet (i.e. after it receives a CTS control packet from its next hope (B) which is interpreted as an confirmation of the transmission success of its CTS), sender (S1) exchanges a DATA/ACK packets with it. From node A, the forwarding process of CTS packets continues between the nodes included in the routing path (nodes B, C, D in Fig. 8) until reaching the Sink. This process rapidly advances towards de Sink and aims to reserve nodes belonging to the path by remaining them active and by putting into sleep mode all corresponding neighbors. The process of exchanging DATA/ACK packets between active nodes advances and follows the reservation process. To each DATA/ACK communication corresponds a transmission time during which each neighbor can be switched into sleep mode for energy saving. For example *t1* is a necessary duration during which node S1 can transmit RTS/CTS/DATA/ACK packets to its next hop A.

The algorithm for CL-MAC, given by Fig. 11, implements at each node these two processes in addition to a wake-up/sleep mechanism required for energy saving purposes. The idea behind a sleeping mechanism on which our protocol CL-MAC is based enables to eliminate all sources of energy waste described in section 1. This makes our protocol suitable to a

ALGORITHM FOR CL-MAC

```
Input :  Table; //Routing table
Begin
Build-a-neighbor-list (Liste) ;
Synchronize-the-schedule;
Get-routing-table(Table) ;   //By recovering routing information stored
                             //by the routing agent
Label1: Nav = 0;
State = WakeUp;
Repeat
If There-is-data-to-send and (State = WakeUp) Then
   If Channel-is-free Then
    If Other-nodes-want-to-access-to-medium Then
         Backoff-Procedure-to-resolve-contention-problem;
    EndIf
     Destination = Get-destination-from-Table(Table);
     Send RTS to Destination;
     State = Wait-for-CTS;
   Else
     Put-into-Sleep-mode-and-WakeUp-next-frame ;   Goto
    Label1 ;
   EndIf
 Else
   If NOT(Channel-is-free) Then
     Receive (message);
     If   (message→destination = ID)   Then   // ID of the receiver
                                               //node of the packet
        Case Type-of(message) Of
          'RTS'   :  Build-and-send (CTS);  State = Wait-for-
                  DATA;
         'DATA' : Build-and-send-to-source (ACK);
                  Rebuild-and-send-DATA-to-next-hop (DATA);
                  State = Wait-for-ACK;
           'ACK' : Put-into-Sleep-mode ;
                  Goto Label1;
           'CTS' : If (State = Wait-for-CTS) Then
                     Send (Data); State = Wait-for-ACK;
                   Else If State ≠ Wait-for-DATA Then
                          Rebuild-and-send (CTS)
                          State = Wait-for-DATA ;
                        EndIf
                   EndIf
        EndCaseOf
      Else
        Nav = message→duration;
        Put-into-Sleep-mode ; Goto Label1;
     EndIf
    Else
     Put-into-Sleep-mode-and-WakeUp-next-frame; Goto Label1;
   EndIf
EndIf
Until (Battery-level < Threshold) ;
Fin CL-MAC
```

Fig. 11. Detailed algorithm for CL-MAC at each sensor node

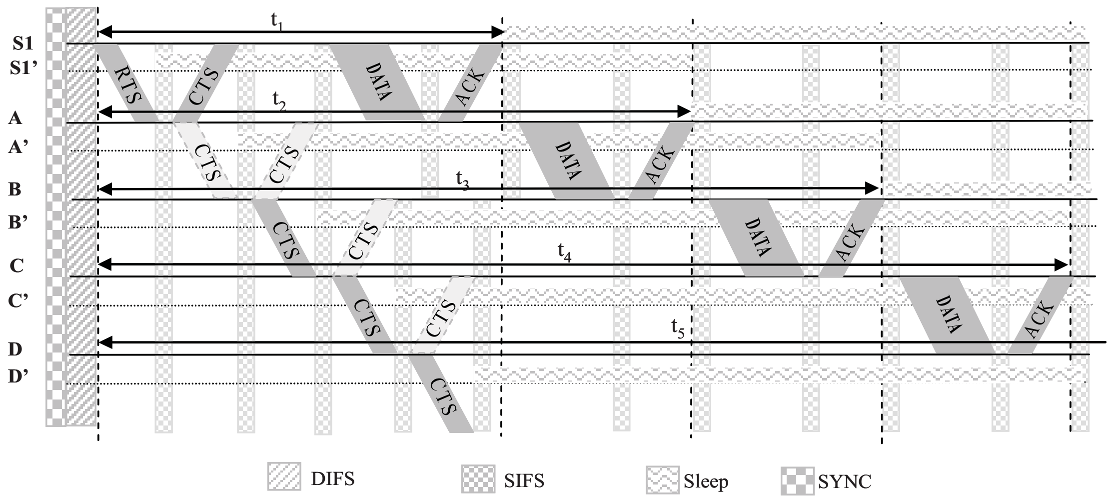

Fig. 12. Temporal behaviour of CL-MAC

very large spectrum of applications where the guaranty of the end-to-end delay transmissions and, at the same time minimum energy expenditure, is very challenging.

Lines 11 and 12 address a case of contention problem when, for example, two RTS control packets collide in case two source nodes want to transmit a DATA packet at the same time. If this problem happens, CL-MAC uses a backoff procedure to resolve this problem.

There are different ways to deal with a sleep schedule in the algorithm:

- Sleep schedule after the reception of an ACK control packet (line 29) resulting from a success of the DATA transmission. After a sleeping period, a node wake-up according to a current sleep schedule.
- Sleep schedule indicated by NAV of a neighbor node not included in the urgent routing path (line 41).
- Finally, sleep schedule of each node not concerned by the urgent communication and that has nothing to send (line 44). In this case, the waking up is planned to be happen according to the current sleep schedule.

Each sensor node executes the CL-MAC algorithm as long as the battery level does not reach a threshold.

3.4 Advantages of CL-MAC

Fig. 13 and 14 illustrate the main advantage of the CL-MAC protocol comparing to others MAC-protocols like S-MAC and MAC-CROSS.

As explained above, in the S-MAC protocol, a frame is divided into two fixed periods: one for a listening which causes a useless loss of energy consumption and another for sleeping. In the listening period, the control messages CTS/RTS are exchanged. The CSMA/CA mechanism is used for packet transmission in order to announce to a source node the next data transmission: for example, source node A in Fig. 13. It is clear that a node like B', not concerned by the communication (between nodes A and B), must enter into sleep mode during all the communication time or after, according to the listening period fixed by the protocol. Just after the end of the communication, node B can starts another communication without waiting until the expiration of sleep period, e.g. waiting for the next frame. This was adopted for designing an adaptive S-MAC protocol. In each frame, the MAC-CROSS

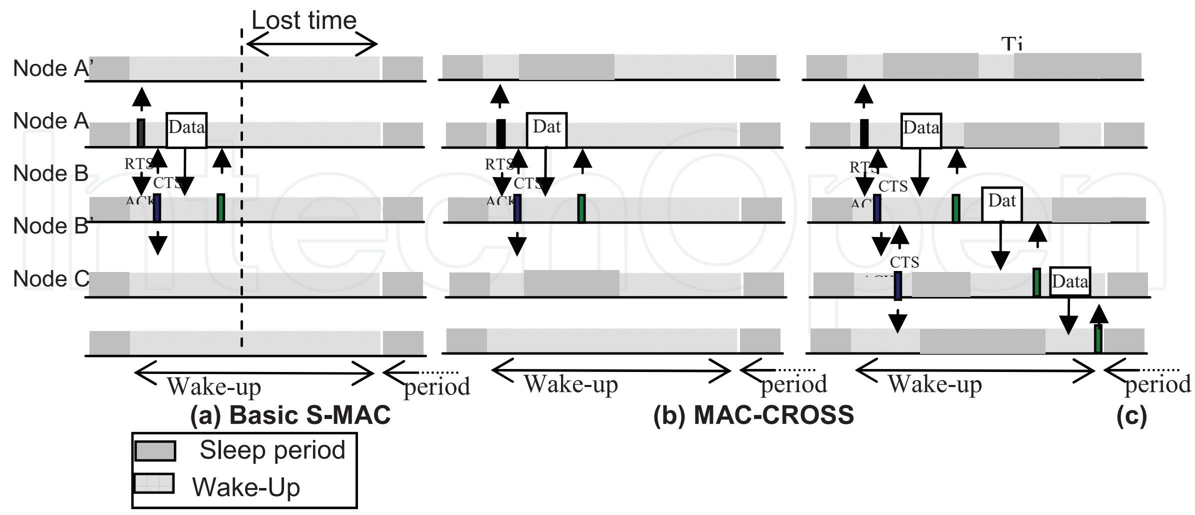

Fig. 13. Main functionalities of S-MAC (a) MAC-CROSS (b) and CL-MAC (c)

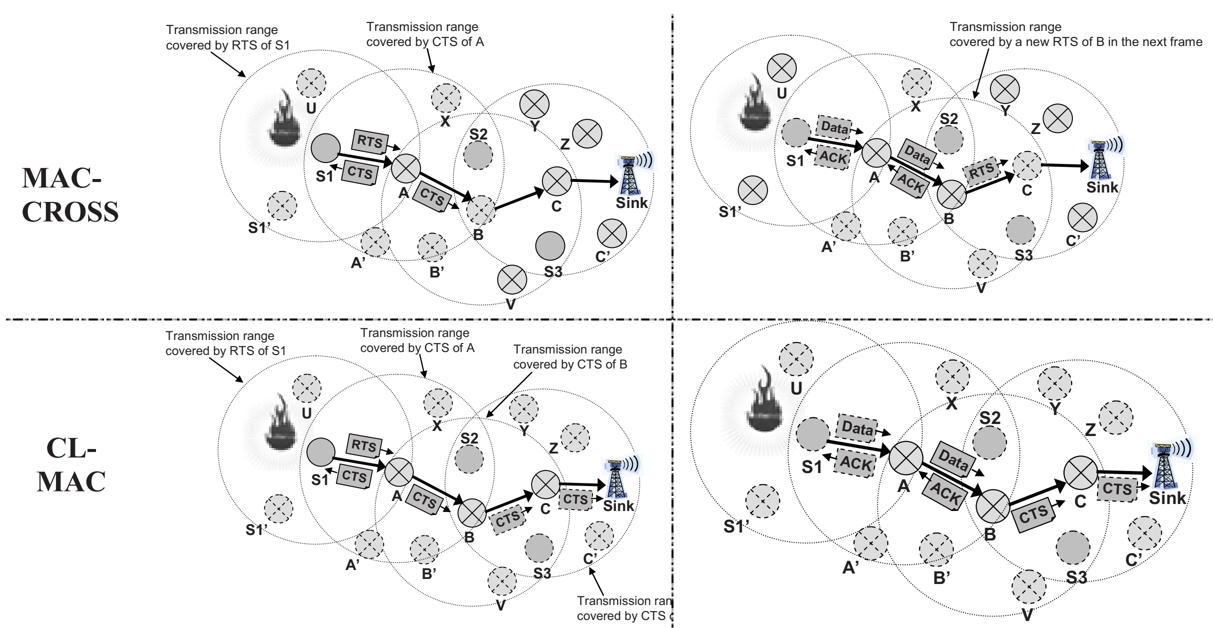

Fig. 14. Main advantages of CL-MAC protocol

protocol enables the exchange of *RTS/CTS/DATA/ACK* messages between three consecutive nodes using routing information obtained from the routing agent in network layer, while the CL-MAC does the exchange of the same messages but only at the beginning of each frame. Then, the third node transmits however directly the received *CTS* to the next node in the path. This *CTS* message is interpreted in case of CL-MAC by a receiver node as a *RTS* of normally the next frame in MAC-CROSS protocol, for each routing node to another in the path until arriving to the Sink during the same frame. This allows all nodes belonging to the routing path to remain active (or reserved) until each node transmits successfully data packets to the next node. Just after transmitting these data packets, the node either enters into sleep mode or prepares to begin a new transmission frame as a new source if it has data to send.

3.5 Other details about CL-MAC

Several aspects of CL-MAC must be studied for showing all of its advantages. Among these aspects, lies the problem of the fragility of the path being reserved for a long time (long frame) and the coexistence of several urgent paths sharing a routing path. We address these two aspects in the following, with illustrations through some scenarios.

Reservation and release of the routing path: Fig. 15 illustrates that the nodes of the routing path reserved for the transmission of data packet (for example nodes S1, A, B) are released when the urgent traffic advances by three hops. They can therefore participate in other communications without waiting until the communication completely finishes between S1 and the Sink. For example, node A' can accept a traffic transmitted by one of its neighbors to relay it to another node, whereas the data packet transmitted by S1 did not reach a Sink yet (it is received by node E in Fig. 15). This shows that the routing path cannot be reserved for long time (long frame) when CL-MAC is applied and therefore the problem of the fragility of the path is consequently excluded.

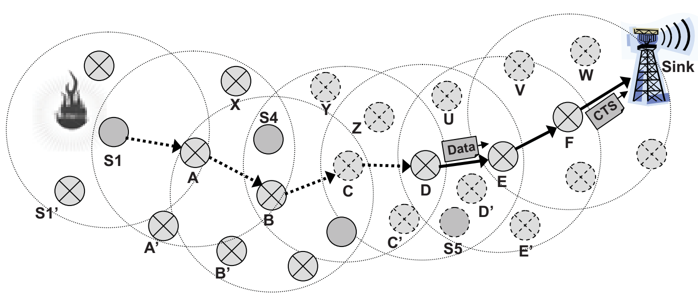

Fig. 15. Reservation and release of the routing path in CL-MAC

*Coexistence of several urgent paths: a*s depicted in Fig. 16, a new urgent path which starts from S5 towards the Sink may include nodes E and F that can be reserved to be used in an another urgent path starting from S1. The reservation of the path S1-A-B-C-D-E-F-Sink (S1-Sink for short) is in progress and before the CTS packet reaches the node E, S5 transmits RTS packet to node E that means it initiates a new communication towards the Sink. In this case, the path S1-Sink may be blocked at the level of node C, because its neighbor D (next hop) can be in sleep mode when node C wishes to communicate with it. This can be accepted in the context of typical applications in which a nearest urgent event can be detected and received before another urgent event far away which was started in the first place. The nearest event (detection of forest fire in our example such as sudden temperature rise) will immediately produce an alarm at the Sink for a fast intervention in the suspected nearest location, but also for dispatching, for example, a team to check the situation in all the sensed area. By doing this, the event far away will be detected even if the urgent message related to it can not be reached by the Sink.

4. Modeling CL-MAC protocol

In order to formally prove the correct operation of our protocol, it is important to model it in

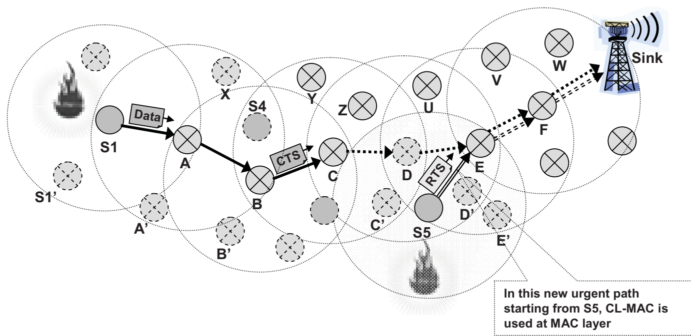

Fig. 16. Case of coexistence of two delay sensitive paths: S1-A-B-C-D-E-F-Sink and S5-E-F-Sink

a convenient mathematical model according to its specifications. The introduction of time Petri nets (Merlin, 1974; Berthomieu & Menasche, 1983) is motivated by their ability to model easily temporal constraints and the existence of a TiNA analyser tool for properties verification. Time Petri nets or TPN for short are a convenient model for real time systems and communication protocols (Berthomieu & Diaz, 1991).

TPN extend Petri nets by associating two values (min, max) of time (temporal interval) to each transition. The value min (min ≥ 0), is the minimal time that must elapse, starting from the time at which transition *t* is enabled until this transition can fire and max (0 ≤ max ≤ ∞), denotes the maximum time during which transition *t* can be enabled without being fired. Times min and max, for transition *t,* are relative to the moment at which *t* is enabled. If transition *t* has been enabled at time α, then *t* cannot fire before α + min and must fire before or at time α + max, unless it is disabled before its firing by the firing of another transition.

Fig. 17 depicts the time Petri net model of CL-MAC protocol. The values in the intervals associated to transitions refers to relative time of transmitting packets (*RTS, CTS, DATA, ACK*) according to IEEE 802.11 standard (IEEE Std, 1999) and respected by our proposal.

4.1 Model hypotheses

The hypotheses on the behaviour of CL-MAC are as follows: we assume that *DIFS* duration = *SIFS* duration = 1 time unit, Control messages *RTS*, *CTS* and *ACK* consume 3 time units, DATA requires 10 time units for its transmission. Initially, only the places *p1*, *p8*, *p14* and *p17* are marked by one token.

4.2 Model explanation

The meanings of the transitions are given in Table 1. In Fig. 17, the sender part, modelling the node wishing to forward a packet to the Sink, is described by four transitions specifying the node behaviour: *t1* and *t2* to send *RTS* and *DATA* packets respectively at precise moments. These moments are indicated by the intervals associated with the transitions (an *RTS* packet is sent after a *DIFS* and *DATA* packet is sent after having received a *CTS* within 3 to 4 units of time, this is represented by the temporal interval associated to transition *t2*). Transition *t3* allows node to enter into sleep mode after having received the *ACK*, whereas *t4* allows the token to remain in the place *p19* (sleep state), time needed before beginning the next communication.

The Receiver part reacts to packets transmitted by the network part. The transition *t5*, generating the *CTS* control packet, is activated once the token, coming from the firing of transition *t1*, takes place in *p2*. It is allowed to fire at the moment $3 \leq \tau \leq 4$ (time for the

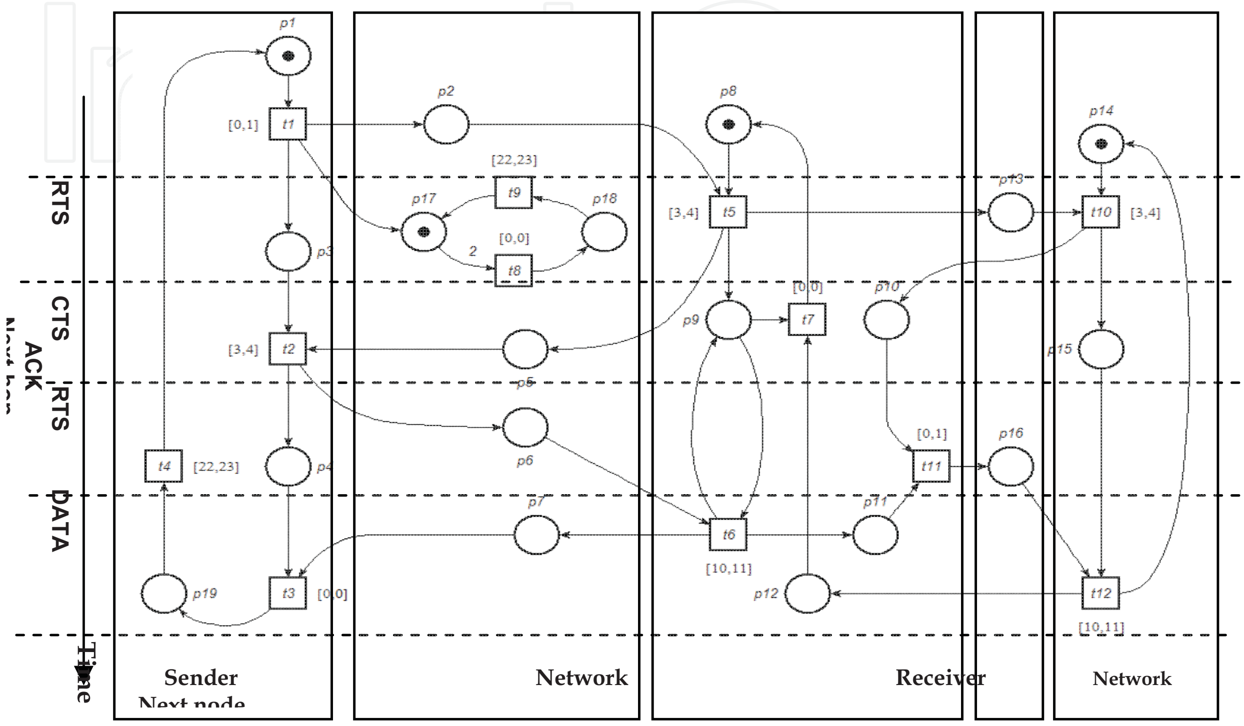

Fig. 17. Time Petri net model for CL-MAC protocol: columns represent temporal behaviour and lines represent spatial behaviour

Transition	Meanings
t1	Transmission of *RTS* packet in unicast mode but the nature of the medium acts as broadcast
t2	Transmission of *DATA* packet in unicast mode
t3	Sender node enters sleep mode
t4	Awakening of the sender to start a new frame
t5	Transmission of *CTS* packet in multicast mode by the receiver
t6	Transmission of *ACK* packet in unicast mode after receiving *DATA* packet
t7	Switching to sleep mode of the receiver after achieving his communication (frame)
t8	Entering sleep mode of neighbor node not concerned by the current communication
t9	Awakening of neighbor node not concerned by the current communication after the last one ends
t10	Transmission of *CTS* packet by the next node
t11	Transmission of *DATA* packet by the receiver to the next node
t12	Entering sleep mode of the next node

Table 1. Model transitions and their corresponding explanation

reception of *RTS* + *SIFS* time*:* 3, 3+1). In the same way, the transition *t6*, models the transmission of an *ACK* control packet after receiving *DATA* packet*.*
The Network part is modelled by two transitions *t8* and *t9* and a set of places *{p2, p5, p6, p7, p17, p18*}. The places *p2*, *p5, p6* and *p7* respectively represent the propagation of *RTS, CTS, DATA and ACK* packets in the wireless medium.
The part formed by the two transitions *t8, t9* and the places *p17*, *p18* models the vicinity of the sender and that of the receiver. A node which wakes up (a token in the place *p17*) and receives a packet which is not intended to it, immediately returns into sleep mode. This explains the labelling of the transition *t8* by the interval [0, 0]. The transition *t9* plays the role of *t4* for the neighbor.

5. Validation of CL-MAC protocol

The TPN proposed for CL-MAC has been validated by the TINA tool software (Berthomieu & Vernadat, 2006; Berthomieu et al., 2004; *http://www.laas.fr/tina*). In this section, a brief presentation of the TINA tool is introduced. The obtained results analysis are also given.

5.1 Presentation of the TINA tool

Tina (TIme Petri Net Analyzer) is a software environment to edit and analyze Petri Nets and Time Petri Nets. It is developed and maintained by a group of researchers of the LAAS/CNRS laboratory at Toulouse University – France (*http://www.laas.fr*). TINA is a powerful tool which allows the checking of many aspects of Petri Nets (bounded, deadlock, re-initialization). It is based on the intrinsic properties of Petri Nets and proposes in particular a functionality of formal validation which brings the mathematical proof that the studied property is checked with a confidence degree of 100%. The functionalities of TINA allow a temporal study of a TPN model based on the reachability analysis method for usual Petri nets. Before using TINA, other software tools in connection with Petri Nets were investigated, in particular HPSim (*http://www.winpesim.de*) and ARP (*http://www.ppgia.pucpr.br/~maziero/doku.php/software:arp_tool*). However, TINA was chosen for its capacities for TPN formal analysis.
Tina accepts input in graphical or textual formats, including PNML (an XML based exchange format for Petri nets).Transition system outputs can be produced for external checkers in a number of textual or binary formats.

5.2 Results analysis

TINA allowed us to validate the following properties of our CL-MAC TPN model:
Boundedness property: this property relates to the finite number of tokens in each place of a TPN for any marking reachable from an initial marking. A TPN is said to be safe if it is 1-bounded. This aspect must be checked in the first place. Otherwise the formal checking of others properties does not have any significance. The not bounded property is characterized by an infinite number of tokens in at least one place among other places of the TPN. In this case, we notice that the model diverges or the implemented system will require an abnormally high quantity of resources (memory, CPU time, etc).
Liveness property: this property allows the detection of portions of died code, i.e. the absence of liveness of some places and/or the blocking of some transitions from TPN, for any initial

and accessible marking from the network. The absence of liveness makes it possible to highlight portions of code which are never performed (thus to detect the modelling errors) and situations where the system modelled is likely to be blocked.

Reversibility property: the system re-initialisation supposes that the system finds its initial state (initial marking) on the basis of any other state during its operation. This property is fundamental to validate automata based systems which present a cyclic operation.

These three properties of our model have been successfully validated by TINA. In the following, we present the classes generated by the TINA tool and the reachability graph formed by these classes. From these results, the validation of these properties has been proved.

*Classes: t*he following classes (*C0* to *C18*) are generated by TiNA tool during the analysis phase:

class 0 p1 p14 p17 p8, 0 <= t1 <= 1
*class 1 p14 p17*2 p2 p3 p8, 3 <= t5 <= 4, 0 <= t8 <= 0*
class 2 p14 p18 p2 p3 p8, 3 <= t5 <= 4, 22 <= t9 <= 23
class 3 p13 p14 p18 p3 p5 p9, 3 <= t2 <= 4, 18 <= t9 <= 20, 3 <= t10 <= 4
class 4 p10 p15 p18 p3 p5 p9, 0 <= t2 <= 1, 14 <= t9 <= 17, t2 - t9 <= ~14
class 5 p10 p15 p18 p4 p6 p9, 10 <= t6 <= 11, 14 <= t9 <= 17
class 6 p10 p11 p15 p18 p4 p7 p9, 0 <= t11 <= 1, 0 <= t3 <= 0, 3 <= t9 <= 7
class 7 p15 p16 p18 p4 p7 p9, 10 <= t12 <= 11, 0 <= t3 <= 0, 3 <= t9 <= 7
class 8 p15 p16 p18 p19 p9, 10 <= t12 <= 11, 22 <= t4 <= 23, 3 <= t9 <= 7
class 9 p15 p16 p17 p19 p9, 3 <= t12 <= 8, 15 <= t4 <= 20, t12 - t4 <= ~11, t4 - t12 <= 13
class 10 p12 p14 p17 p19 p9, 11 <= t4 <= 13, 0 <= t7 <= 0
class 11 p14 p17 p19 p8, 11 <= t4 <= 13
class 12 p10 p11 p15 p18 p19 p9, 0 <= t11 <= 1, 22 <= t4 <= 23, 3 <= t9 <= 7
class 13 p15 p16 p18 p19 p9, 10 <= t12 <= 11, 21 <= t4 <= 23, 2 <= t9 <= 7, t4 - t9 <= 20, t9 - t4 <= ~15
class 14 p15 p16 p17 p19 p9, 3 <= t12 <= 9, 15 <= t4 <= 20, t12 - t4 <= ~10, t4 - t12 <= 13
class 15 p12 p14 p17 p19 p9, 10 <= t4 <= 13, 0 <= t7 <= 0
class 16 p14 p17 p19 p8, 10 <= t4 <= 13
class 17 p13 p14 p18 p4 p6 p9, 0 <= t10 <= 1, 10 <= t6 <= 11, 14 <= t9 <= 17, t10 - t9 <= ~14
class 18 p10 p15 p18 p4 p6 p9, 9 <= t6 <= 11, 14 <= t9 <= 17, t9 - t6 <= 7

These classes represent the chronological temporal behavior of CL-MAC by using the firing mechanism of the corresponding time Petri net.

- *Reachability graph :* Fig. 18 depicts the reachability class graph of time Petri net representing the previous classes. This analysis reveals that the CL-MAC protocol has effectively the good properties previously mentioned:

- *Liveness*: the net is deadlock freeness and each transition is always able to be fire infinitely.
- *Boundedness*: the number of tokens in every place is limited to one token except the place *p17* that is 2-bounded.
- *Reversibility*: the return of the TPN to its initial state shows that the CL-MAC TPN model is reversible (able to be reinitialized).

Fig. 19 summarizes some results given by *Net Draw Control* of the TiNA tool.

6. Evaluation of the CL-MAC protocol

Performance evaluation of our protocol includes some metrics such as energy and latency which must be provided in some typical applications as metrics of performance. Because of the lack in practice of simulation tools dedicated to Cross-layering design approach based protocols, we have developed our home simulator software using the C++ Builder programming language. Using this software, we have implemented and compared the CL- MAC protocol described by the algorithm given by Fig. 11 with the MAC-CROSS and S-

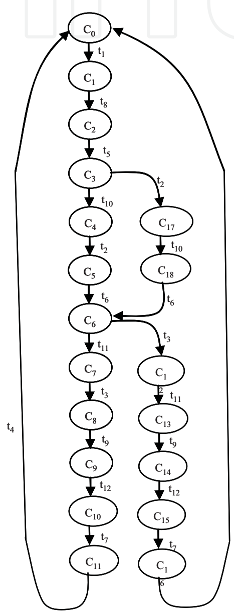

Fig. 18. Reachability class graph

LIVENESS ANALYSIS
……………………………………………………………………………
Possibly live
0 dead classe(s), 19 live classe(s)
0 dead transition(s), 12 live transition(s)
STRONG CONNECTED COMPONENTS:
0: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
SCC GRAPH:
0 -> t1/0, t8/0, t5/0, t10/0, t2/0, t6/0, t11/0, t3/0, t9/0, t12/0, t7/0, t4/0
0.000s

Fig. 19. Analysis results given by TiNA tool

MAC protocols presented above. This comparison is justified by the fact that : in one hand MAC-CROSS is considered as a basis of our work and in the other hand S-MAC is regarded as a reference chosen by the community of researchers for studying energy efficient MAC layer issues for WSN. All experimentations have been performed according to the scenario given by Fig. 9 in which one or several sources transmit delay sensitive traffic towards the Sink.

6.1 Simulation Environment

An example of deployed WSN network generated by our simulator and used for CL-MAC evaluation is illustrated by Fig. 20 and simulation parameters are summarized in Table 2. The latency (μ second) used in our experimentations is defined as an elapsed time between the time of message sending by a source node and the time of arrival of this message to the final destination (Sink). In order to compute the energy consumed during each data transmission from the source node to the Sink, we have used a first order energy model introduced by Heinzelman et al. (2000).

6.2 Performances analysis

The aim of simulation is to analyze the effect of the variation of some parameters such as the number of data sources, density and hop number on CL-MAC, MAC-CROSS and S-MAC behaviour in terms of dissipated total energy and latency in the network. Fig. 21 shows that if the number of data sources increases in the network, the total energy consumed by MAC-CROSS and S-MAC increases more quickly than that consumed by CL-MAC. But, more than 43 active data sources simultaneously relative to MAC-CROSS and 46 relative to S-MAC, make our protocol consuming more energy. This can be explained by the fact that each time a data source is added to the network, more nodes will be mobilized to remain in

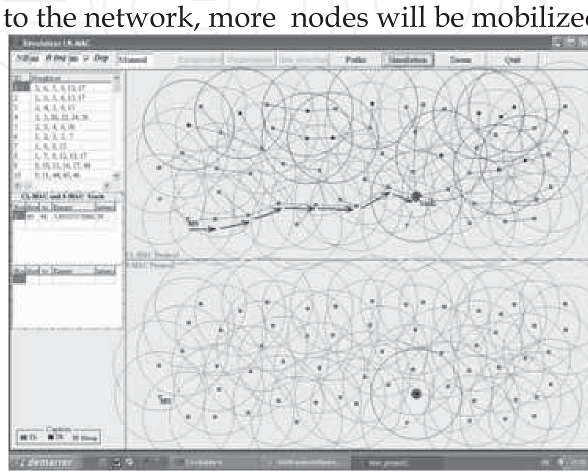

Fig. 20. Example of WSN with 60 sensor nodes. This simulation interface snapshot is divided into two windows: the top window for CL-MAC and the bottom window for S-MAC, the left part of the screen is reserved for simulation traces.

Parameter Type	Test Value
Number of sensor nodes (in the same deployment space)	Changes according to the evaluation example
Number of Sink	01
RTS/CTS Message size	118 bits
ACK message size	112 bits
Data message size	800 bits
Throughput	8 bits/µs

Table 2. Simulation parameters

active state (thus their duty cycle increases) in order to participate in routing paths as intermediate nodes. Thus, there will be fewer nodes in sleep state (but the effectiveness of our protocol is based on its ability to put into sleep mode any node in the vicinity not concerned by the routing operation). On the other hand, S-MAC has the capability to put into sleep mode each sensor node for a half cycle independently of the number of source nodes. Therefore, no changes will be made in the node's duty cycle. MAC-CROSS behaves like S-MAC when traffic loads are heavy, except that MAC-CROSS begins a new frame after each two successive hops.

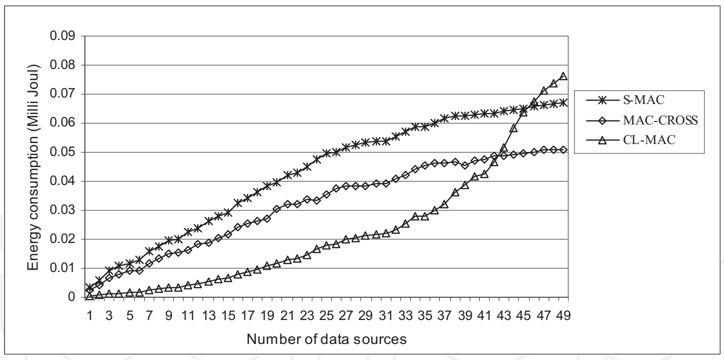

Fig. 21. Energy consumption vs the number of data sources with 60 deployed nodes

In the experimentation producing the results of Fig. 22, we have varied the density of the network in a fixed deployment space and we have chosen a delay sensitive data source located at 10 hops far from the Sink. The data source sends data packets to the Sink on the routing path. Because the number of neighbors of each node belonging to the path is variable, we have chosen the average number of these neighbours in numerical evaluation of total energy consumed by the network. In Fig. 22, CL-MAC always maintains its best performance as a protocol with a minimum energy consumption compared to MAC-CROSS and S-MAC. The latter consume more energy if the network size increases.

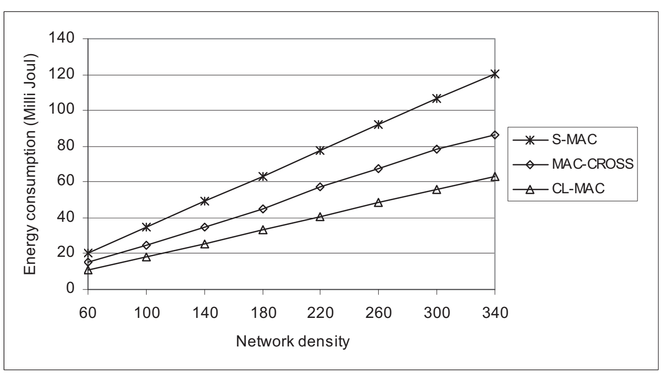

Fig. 22. Energy consumption in case of varying the network density

Fig. 23 shows well that latency increases quickly in the case of CL-MAC when a new source node of urgent traffic is added to the number of nodes already in communication. On the other hand we note, for a given number of sources lower than 3 relative to MAC-CROSS and 4 relative to S-MAC, that CL-MAC records latency was better than MAC-ROSS's and S-MAC's. This can be explained by the nature of the progressively generated urgent paths. Indeed, in the presence of not disjoint paths (paths having at least one intermediate node in common), the CL-MAC protocol leaves in active mode only one node included in a routing path and puts the remainder of neighbor nodes into sleep mode. However, if this active node is included in several routing paths, then the only path, which will be operational, is the one that will have possession of this active node. The other paths will be delayed until the current routing step at this active node is finished. More precisely, the neighboring nodes of the active node, that have been included in other routing paths, will be found in sleep mode at the moment when the nodes, which precede them in their

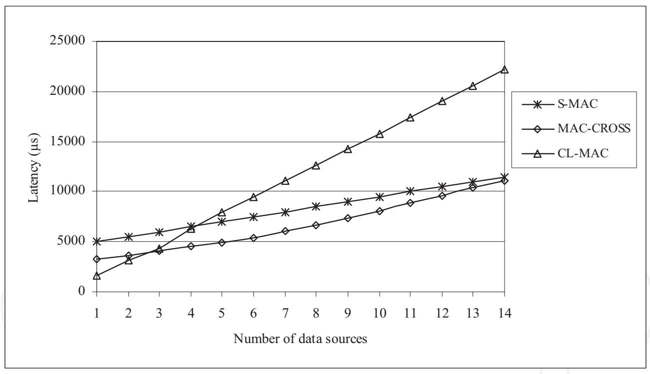

Fig. 23. Latency vs number of data sources with 150 deployed nodes and hop number=10

corresponding paths, want to transmit a message to them. It is consequently clear that in the case of disjoint paths that have no common neighbors, CL-MAC maintains its performances. However, the guaranty of the presence of these disjoint paths in the network from many sources is in practice a very challenging issue. We can see again that MAC-CROSS converge to S-MAC when this number of data sources is greater than 14. This is due, as we said previously, to the fact that when traffic loads are heavy, MAC-CROSS behaves like S-MAC.

The results of Fig. 24 are obtained by carrying out 5 transmissions of data packets for each hop where the resulting average value is considered. We note in this figure that each time a sensor node moves away from the Sink, the CL-MAC protocol records a better latency compared to MAC-CROSS and S-MAC. Indeed, while moving away from the Sink, the routing path will contain more sensor nodes participating in the routing operation. Therefore, according to the strategy on which is founded CL-MAC, there will be more potential sensor nodes able to switch into sleep mode. This explains the gain in term of latency in presence obviously of only one routing path at a given time.

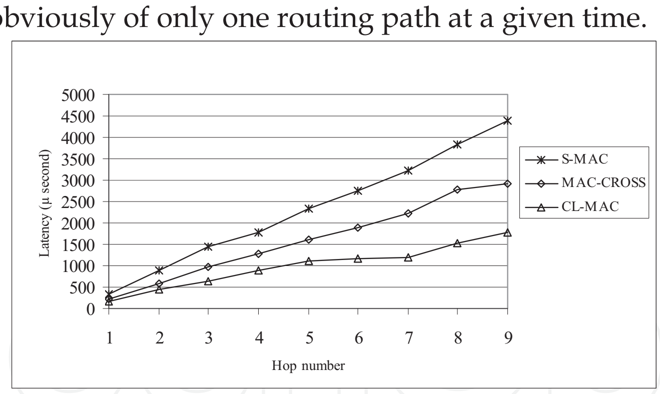

Fig. 24. Latency vs hop number with 60 deployed nodes

7. Conclusion and future work

In this chapter, we have proposed a novel energy efficient and low latency MAC protocol for delay sensitive traffic in WSN using a Cross layer optimization approach, named CL-MAC. In order to prove the correctness of its operation, we have modelled it using time Petri nets and some related properties like liveness, boundedness and reversibility have been validated analytically using the TiNA tool. Various experiments that we have performed show that CL-MAC outperforms MAC-CROSS and S-MAC in term of energy saving and low latency in the following cases: 1) when the data source is far from the Sink, and 2) when, at a given time, the number of data sources is low. However, CL-MAC described in this chapter acts to enable energy saving and low latency in each urgent path only. But in default operation mode (absence of delay sensitive traffic) nodes must use an adequate MAC protocol to deliver a normal traffic (for example the ambient temperature measured periodically) from sources to the Sink using best effort paths. CL-MAC must be extended to take into account this second kind of traffic. We propose as future work an Adaptive Cross-Layer MAC protocol (ACL-MAC) that could be able to differentiate and manage these two types of traffic.

8. References

Akyildiz, I. F. & Ismail, H. K. (2004) 'Wireless sensor and actor networks: research challenges', Georgia Institute of Technology, *www.sciencedirect.com*.

Akyildiz, I. F.; Vuran, M. C. and Akan, O. B. (2006) 'A Cross layer protocol for wireless sensor networks', *Proc. Conference on Information Sciences and Systems (CISS'06)*, Princeton, NJ.

Bachir, A.; Barthel, D., Heuss, M., Duda, A. (2006) 'Micro-Frame Preamble MAC for Multihop Wireless Sensor Networks', IEEE ICC2006 Proceedings.

Berthomieu, B.; Vernadat, F. (2006) 'Time Petri Nets Analysis with TINA', tool paper, *In Proceedings of 3rd Int. Conf. on The Quantitative Evaluation of Systems (QEST 2006), IEEE Computer Society.*

Berthomieu, B.; Ribet, P.O., Vernadat, F. (2004) 'The tool TINA - Construction of Abstract State Spaces for Petri Nets and Time Petri Nets', *International Journal of Production Research,* Vol. 42, no 14.

Berthomieu, B. & Diaz, M. (1991) 'Modeling and Verification of Time Dependent Systems Using Time Petri Nets', *IEEE Trans. On Soft. Eng.*,Vol. 17, n° 3, pp. 259-273.

Berthomieu, B. & Menasche, M. (1983) 'An Enumerative Approach for Analyzing Time Petri Nets', *IFIP Congress Series*,Vol. 9, pp. 41-46 North Holland.

Buettner, M.; Gary Y., Eric, A. and Richard, H. (2006) 'X-MAC: A Short Preamble MAC Protocol For Duty-Cycled Wireless Sensor Networks', *Proceedings of the 4th international conference on Embedded networked sensor systems*, pp.307-320.

Cheng, H.; Qin, L. and Xiaohua, J. (2006) 'Heuristic Algorithms for Real-time Data Aggregation in Wireless Sensor Networks', School of Computing, Wuhan University, China. *IWCMC'06*, July 3–6.

Demirkol, I.; Ersoy, C., Alagoz, F. (2006) 'MAC protocols for Wireless Sensor Networks: a survey', *IEEE Communications Magazine*, pp.115-121.

Dewasurenda, D.; Mishra, A. (2005) 'Design Challenges in Energy-Efficient Medium Access Control for Wireless Sensor Networks', *Handbook of Sensors Networks: Compact Wireless and Wired Sensing Systems*, Edited by Mohammad Ilyas and Imad Mahgoub, CRC Press.

EL-Hoiydi, A. (2002) 'Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks', *In IEEE International Conference on Communications (ICC)*, New York.

Finn, G. G. (1987) 'Routing and addressing in large metropolitan-scale internetworks', *ISI Research Report*, ISU/RR-87–180.

Halkes, G.P.; Van Dam, T., Langendoen, K.G. (2005) 'Comparing Energy Saving MAC Protocols for Wireless Sensor Networks', *Mobile Networks and Applications 10*, pp.783-791, Springer Science.

Heinzelman, W. R.; Chandrakasan, A. P., Balakrishnan, H. (2002) 'An application-specific protocol architecture for wireless microsensor networks'. *IEEE Transactions on Wireless Comm.*, 1 (4): 660–670.

Heinzelman, W. R.; Chandrakasan, A., Balakrishnan, H. (2000) 'Energy-Efficient Communication Protocol for Wireless Microsensor Networks', *The Hawaii International Conference On System Sciences*, January 4-7, MAUI, HAWAII.

Holger, K.; Marc, L. and Tim, N. (2003) 'A Data Aggregation Framework for Wireless Sensor Networks', *European research project EYES*, Berlin.

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

IEEE Std. 802.11- (1999), *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, ISO/IEC 8802-11:1999(E), IEEE Std. 802.11.

Ignatius, M. (2006) 'Energy-efficient Wireless Sensor Network MAC Protocol', *PhD Thesis in Electrical Engineering*, March 31, Faculty of Virginia Polytechnic Institute and State University.

Injong, R.; Ajit, W., Mahesh, A. and Jeongki, M. (2005) 'Z-MAC: an Hybrid MAC for Wireless Sensor Networks', *Proceedings of the 3rd international conference on Embedded networked sensor systems*, San Diego, California, USA, pp.90-101.

Karn, P. (1990) 'MACA - A new channel access method for packet radio'. In ARRL/CRRL Amateur Radio 9th Computer Networking, pp. 134–140.

Koen, L. & Gertjan, H. (2004) 'Energy-Efficient Medium Access Control', The embedded Systems Handbook, Delft University of Technology, Faculty of Electrical Engineering, *Mathematics and Computer Science Mekelweg 4*, 2628CD Delft, The Netherlands.

Lu, G.; Krishnamachari, B. and Raghavendra, C.. (2004) 'An adaptive energy-efficient and low-latency MAC for data gathering in sensor networks', *In Int. Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN),* Santa Fe, NM.

Merlin, P. M. (1974) 'A Study of the Recoverability Systems of Computing', Irvine Univ. California, *PhD thesis*.

Muneeb, A.; Umar, S., Adam, D., Thiemo, V., Kay, R., Koen, L., Joseph, P., Zartash, A. U. (2006) 'Medium Access Control Issues in Sensor Networks', *ACM SIGCOMM Computer Communication Review*, Vol.36.No.2.

Polastre, J.; Hill, J. and Culler, D. (2004) 'Versatile low power media access for wireless sensor networks', in *SenSys'04*, pp.12-24.

Rajendran, V.; Obraczka, K., Garcia-Luna-Aceves, J. J. (2003) 'Energy-efficient, collision-free medium access control for wireless sensor networks', *In: Proc. ACM SenSys 03*, Los Angeles, California.

Sohraby, K.; Minoli, D., Znati, T. (2007) 'Medium Access Control Protocols for Wireless Sensor Networks', *Book chapter in Wireless Sensor Networks : Technology, Protocols and Applications*, Wiley-Interscience, pp.142-173.

Sohrabi, K.; Gao, J., Ailawadhi, V., Pottie G. J. (2000) 'Protocols for Self-Organization of a Wireless Sensor Network', *IEEE Personal Communications*, Vol. 7, No. 5, pp. 16–27.

Sohrabi, K.; Pottie, G. J. (1999) 'Performance of a Novel Self- organization Protocol for Wireless Ad Hoc Sensor Networks', *Proceedings of the IEEE 50th Vehicular Technology Conference (VTC'99)*, pp. 1222–1226.

Suh., C.; Young-Bae, K. and Dong-Min, S. (2006) 'An Energy Efficient Cross-Layer MAC Protocol for Wireless Sensor Networks', Graduate School of Information and Communication', Ajou University, Republic of Korea, *APWeb 2006*, *LNCS 3842*, pp. 410-419.

Wei, Y.; Heidemann, J., Estrin, D. (2002) 'An energy-efficient MAC protocol for wireless sensor networks', Proceedings *of INFOCOM 2002, Twenty-First Annual Conference of the IEEE Computer and Communications Societies*,. IEEE.

Wei, Y.; Heidemann, J., Estrin, D. (2004) 'Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks', *IEEE/ACM Transactions on Networking*, Vol. 12, No. 3.

Willig, A. (2006) 'Wireless Sensor Networks: concept, challenges and approaches', *Elektrotechnik & Informationstechnik*, pp.224-231.

Zhi-Wen, O.; Shruthi, B. K. and Sang S. K. (2005) 'Medium Access Control for Wireless Sensor Networks', *CS258 - Advanced Communication Networks*, San Jose State University.

Petri Nets Applications
Edited by Pawel Pawlewski

ISBN 978-953-307-047-6
Hard cover, 752 pages
Publisher InTech
Published online 01, February, 2010
Published in print edition February, 2010

Petri Nets are graphical and mathematical tool used in many different science domains. Their characteristic features are the intuitive graphical modeling language and advanced formal analysis method. The concurrence of performed actions is the natural phenomenon due to which Petri Nets are perceived as mathematical tool for modeling concurrent systems. The nets whose model was extended with the time model can be applied in modeling real-time systems. Petri Nets were introduced in the doctoral dissertation by K.A. Petri, titled "„Kommunikation mit Automaten" and published in 1962 by University of Bonn. During more than 40 years of development of this theory, many different classes were formed and the scope of applications was extended. Depending on particular needs, the net definition was changed and adjusted to the considered problem. The unusual "flexibility" of this theory makes it possible to introduce all these modifications. Owing to varied currently known net classes, it is relatively easy to find a proper class for the specific application. The present monograph shows the whole spectrum of Petri Nets applications, from classic applications (to which the theory is specially dedicated) like computer science and control systems, through fault diagnosis, manufacturing, power systems, traffic systems, transport and down to Web applications. At the same time, the publication describes the diversity of investigations performed with use of Petri Nets in science centers all over the world.