

# FTA-MAC: Fast Traffic Adaptive Energy Efficient MAC Protocol for Wireless Sensor Networks

Van-Thiep Nguyen, Matthieu Gautier<sup>(✉)</sup>, and Olivier Berder

IRISA, University of Rennes 1, Lannion, France  
{van-thiep.nguyen,matthieu.gautier,olivier.berder}@irisa.fr

**Abstract.** This paper presents the FTA-MAC (Fast Traffic Adaptive MAC) protocol, a novel energy-efficient MAC protocol based on asynchronous duty cycling for Wireless Sensor Networks (WSNs). In FTA-MAC protocol, the communication between the sensor nodes is initiated by sending a wake-up beacon from the receiving node. The latter fast adapts its *wake-up interval* according to the traffic rate in order to reduce the *idle listening* of the transmitting nodes, which results in reducing the energy consumption. FTA-MAC protocol is implemented and evaluated in OMNeT++/Mixim network simulator. Simulation results show that FTA-MAC outperforms state-of-the-art protocols under fixed and variable traffic rates and also with multiple concurrent nodes. Thanks to this better wake-up schedule, the lifetime of a sensor node is increased 1.5 to 2 times.

**Keywords:** Energy-efficient MAC protocol · Traffic aware · Wireless sensor networks · Wake-up interval

## 1 Introduction

Wireless sensor networks (WSNs) received tremendous attention in recent years because of the evolution of sensor devices as well as wireless communication technologies. The sensor devices connected to the Internet of Things (IoT) [1] can be used to enable remote health monitoring, emergency notification or other observation systems. A problem of such sensor devices is the limited lifetime because of the limitation of the battery capacity. To tackle this problem, one solution is to use an energy efficient protocol to control the behavior of each device in the network. Another challenge in these systems is to deal with variable traffic that can appear in applications such as remote health monitoring. In this type of systems, the sensor nodes are used to monitor the patient physiological parameters such as pulse rate, body temperature, etc., which are periodically transmitted to the center node (e.g. a smart phone). However when the observed patient changes its activity, the sensor nodes can also modify their data rate. Therefore, a new communication management mechanisms are required allowing these devices to adapt with respect to the variable traffic.

The Medium Access Control (MAC) layer allows a sensor node to efficiently share the wireless medium with others in the network. In this layer, the main causes of energy consumption are the idle listening, the overheads, the overhearing and the collisions [2]. Thus, in order to reduce the energy, these factors need to be optimized but there are trade-offs between them. For example, reducing idle listening and collisions requires extra synchronization and overheads, whereas reducing the synchronization and overheads causes the waste of energy in collisions. In the context of energy-efficient MAC protocols, an important mechanism for reducing the energy consumption is duty cycling. In this technique, the radio is turned on periodically, switching between awake and sleeping states.

The recent duty cycling MAC protocols can be grouped into two types: synchronous and asynchronous. While synchronous duty cycling MAC protocols (such as SMAC [3], TMAC [4]) reduce the energy consumption by synchronizing the sleep and wake-up times of sensor nodes, asynchronous duty cycling protocols do not require any synchronization period and can be categorized into two groups: sender initiated (e.g. BMAC [5], XMAC [6], WiseMAC [7]) and receiver initiated (PW-MAC [8], RICER [9], RI-MAC [10], TAD-MAC [11]). In asynchronous protocols, the receiver/sender sends a Wake-up Beacon (*WB*) to schedule a rendez-vous with other nodes for receiving/sending data frames.

While many of these protocols have been proposed to reduce energy consumption, few of them address the case of WSNs with variable traffics. Indeed, the traffic in WSNs can be variable depending on the requirements of the applications. The senders can change their traffic rate many times and without any preview.

In this paper, a new asynchronous duty cycling energy efficient MAC protocol, called Fast Traffic Adaptive MAC (FTA-MAC) protocol, is presented. Typically, in a duty cycling protocol, the optimal protocol has the schedule such that the sender and receiver wake up at same time, as it is the case for synchronous protocols but at the price of less flexibility and extra overheads, leading to energy waste. To approach this optimal schedule, FTA-MAC protocol adapts the wake-up interval of the receiver toward the transmission rate of the sender. To deal with variable traffic, FTA-MAC provides an adaptive mechanism at the receiver side to quickly reach a steady state value of wake-up interval according to the new traffic rate of sender.

The contributions of our work are:

- designing of a novel energy-efficient MAC (FTA-MAC) protocol for WSNs with variable traffics. FTA-MAC minimizes the energy consumption of the sensor nodes by reducing their idle listening time.
- implementing FTA-MAC in OMNeT++/Mixim network simulator for a star network topology. The performance of FTA-MAC are compared with TAD-MAC and RICER under fixed and variable traffic rates.

The rest of this paper is organized as follows. Section 2 describes the related works on receiver-initiated and adaptive MAC protocols. The main contribution of the paper is presented in Sects. 3 and 4, which describe the design of

FTA-MAC protocol and its evaluation through network simulations, respectively. Finally, the paper ends with the conclusion in Sect. 5.

## 2 Related Works

One of the early MAC protocols based on duty cycling technique with initiation by the receiver is RICER proposed in [9]. A receiving node wakes up with a constant period  $I_{WU}$  and initiates the communication by transmitting a short *WB* (with a duration  $T_b$ ) and by monitoring the channel for a fixed duration  $T_l$ . If there is no response, this node goes back to sleep. A source node with data to transmit wakes up, stays awake and waits for the *WB* from the destination. Once receiving a *WB*, it sends its data frame. The communication ends with an acknowledgement (*ACK*) sends from the destination after correctly receiving the data frame. Two other versions of RICER were proposed in [12] to avoid data collisions.

The receiver-initiated mechanism has been also applied to asynchronous duty cycling in RI-MAC (Receiver-Initiated MAC) [10]. In RI-MAC, the *WB* is used to initiate the communication. In some cases, the receiver is already active and sends *WB* before the sender wakes up so the latter misses it. An mechanism called *beacon-on-request* aims to avoid this bad situation. The sender sends a special *WB* which contains the receiver address to specify who is supposed to receive the following *DATA* frame. The receiver answers a beacon to enable the sender to send *DATA* immediately. RI-MAC works with dynamic traffic load by increasing or decreasing the number of nodes but does not work for the variable traffic.

Proposed in [8], PW-MAC (Predictive-Wakeup MAC) aims at optimizing the idle listening. PW-MAC schedules the rendez-vous for a receiving node and a transmitting node by using a pseudo-random number generator in each node. The receiver includes the parameters of its generator in the *WB*. Based on these values, the sender can predict the next wake-up of the receiver so that the sender can wake up shortly before the receiver. Although this mechanism reduces the idle listening and energy, it is not suitable for networks with variable traffic. If the sender needs to change its traffic rate, it must buffer the data frame and waits until the receiver wakes up. The traffic latency will increase and frames will be dropped if the buffer is full.

One protocol that includes a mechanism for variable traffic is TAD-MAC (Traffic-Aware Dynamic MAC) protocol, presented in [11]. In TAD-MAC, the receiver periodically sends a *WB* and considers the variable traffic by adapting its wake-up interval to the traffic rate of each individual sender. In order to be reactive to the traffic change, the receiver wakes up twice more frequently than the sender. This leads to energy waste due to unnecessary wake-ups of the receiver. Unlike in RI-MAC, the *WB* of TAD-MAC contains a specific node address to identify the destination address. This mechanism allows the reduction of wireless collisions.

FTA-MAC and TAD-MAC protocols are both based on the Traffic Status Register (*TSR*) technique introduced in the next section. However, FTA-MAC

proposes an algorithm which allows the receivers to estimate the data rate of senders and to wake up just in time to reduce the *idle listening* of senders.

### 3 Fast Traffic Adaptive MAC Protocol

#### 3.1 Traffic Status Register (TSR) Based Protocol

In a network under variable traffic, the traffic estimation is an important part. With their limited computation capacity, the sensor nodes require a light-weight mechanism to estimate the traffic. The *TSR* technique is first introduced in [11] to store the status information of the traffic. When a receiver wakes up to send a *WB*, if it receives a data response, the *TSR* corresponding to the transmitted node is filled with a BIT 1, in contrast, the *TSR* is filled with a BIT 0. The new status (either BIT 1 or BIT 0) is inserted into the *TSR* at the first index by shifting left one bit. In a star network, the receiver contains a list of *TSR* to track the traffic status of all the potential senders, as described in Fig. 1. This is a blind way to keep the track of traffic load without any information about the data rate of the transmitter. This blind estimation is not a quantitative evaluation of the data rate but rather a relative measure of either the traffic is increasing (which is represented by multiple consecutive ones in the *TSR*) or decreasing (which is represented by multiple zeros in the *TSR*). In this context, the sequence [10101010...] seems the best trade-off between the optimal sequence [11111111...] (*i.e.* each wake-up of the node is followed with a successful data reception), and the too frequent wake-up [10001000...] due to several reasons:

- Sequence of [11111111...] is an ideal sequence which means that each time the receiver wakes up, it receives data. Because of the blind estimation, it always has probability of missing data as soon as the traffic rate increases (at the transmitter side). The wake-up interval of the receiver will remain at lower traffic rate.

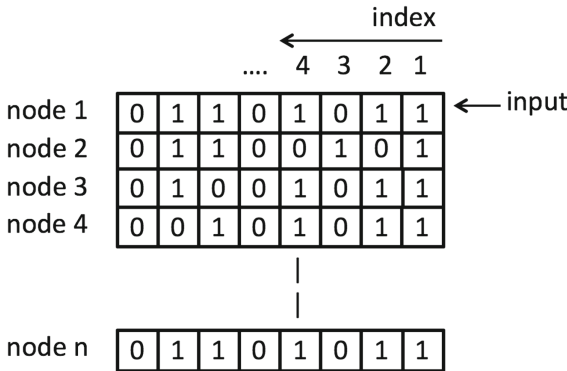
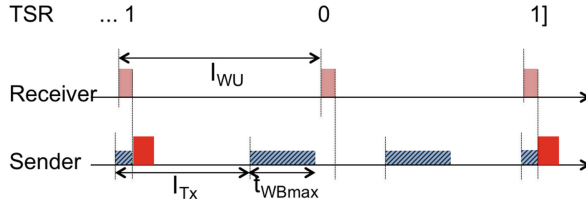


Fig. 1. Traffic Register Status (*TSR*) structure.

- Sequence of [10001000...] is another extreme as the receiver wakes up too often and wastes energy by sending unnecessary *WB*.

Thus, [10101010...] sequence is selected as a steady state in TAD-MAC protocol. The sender periodically wakes up to send data. This period is called interval of transmission ( $I_{Tx}$ ) and the receiver adapts its wake-up interval ( $I_{WU}$ ) according to the  $I_{Tx}$  of sender. With the sequence [10101010...], the receiver wakes up twice more frequently than the sender. It means that  $I_{WU}$  is equal to  $0.5I_{Tx}$ . The sender wakes up and waits for *WB* from the receiver in a limited interval of time ( $t_{WBmax}$ ) and the  $t_{WBmax}$  is always less than  $I_{Tx}$ . Because of the blind estimation, a drawback of the *TSR* technique is that the receiver can converge to a wrong value even if the sequence is still the best trade-off sequence [10101010...] with the condition that  $t_{WBmax}$  is less than  $0.5I_{Tx}$ . This problem is described more clearly in Fig. 2 where the receiver converges to a wrong value:  $I_{WU}$  is equal to  $1.5I_{Tx}$ .

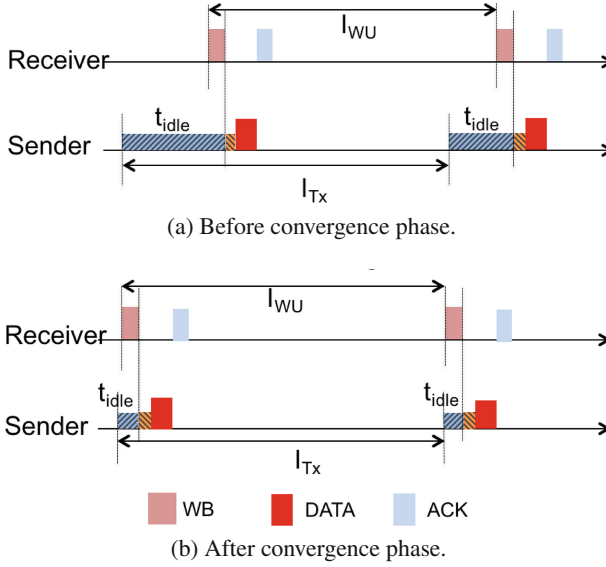
Moreover, the speed of convergence depends on the length of the *TSR* and the pair values of  $I_{WU_0}$  (the initial value of  $I_{WU}$ ) and  $I_{Tx}$ . In some worst case, the receiver takes a long time to reach the steady state and loses much energy. The blind estimation without any additional information from the sender nodes is a good idea but it still has many limitations.



**Fig. 2.** Wrong convergence of TAD-MAC protocol ( $I_{WU}$  converges to  $\frac{3}{2} I_{Tx}$  instead of  $\frac{1}{2} I_{Tx}$ ).

### 3.2 FTA-MAC Protocol Design

FTA-MAC is a novel protocol based on the duty cycling technique. The communications between the sensor nodes are initiated by the receiving node and the principle is described in Fig. 3. The figure is divided into two phases: before and after the convergence. On the one hand, the receiver periodically sends a *WB* frame to notify its wake-up to the other nodes. On the other hand, the sender periodically wakes up with a period  $I_{Tx}$  which depends on its data rate. Before sending the data, it waits for the *WB* from the receiver. This period is called *idle listening* time ( $t_{idle}$ ) which is the most energy consumption activity in receiver-initiated MAC protocols. Once receiving the *WB*, the sender sends the data after sensing the medium channel during an interval of time called  $t_{CCA}$  (Clear Channel Assessment). The communication session ends with an acknowledgement (*ACK*) message from the receiver to the sender after successfully receiving the data frame.



**Fig. 3.** Behaviour of FTA-MAC protocol.

To reduce  $t_{idle}$ , the receiver attempts to wake up according to the  $I_{Tx}$  of each sender. The wake-up interval ( $I_{WU}$ ) of the receiver is adapted itself to the data transmission rate of each transmitting node by collecting the statistics of its data traffic. To this aim, a *TSR* is used to store the recent history of data transmission, as for TAD-MAC protocol.

The second phase after convergence indicates that the receiver has adapted its  $I_{WU}$  to the  $I_{Tx}$  of the senders in such a way that the idle listening time is reduced. This phase is described in Fig. 3b, where the  $t_{idle}$  of the source node is reduced. The adaptive mechanism of  $I_{WU}$  is explained in Fig. 4 and  $I_{WU}$  is calculated by the following adaptive function:

$$I_{WU}(i+1) = \begin{cases} I_{WU}(i) + n_0(i) \cdot t_{ref} & \text{if } TSR(i) = 0 \\ \frac{\sum_{j=k}^i I_{WU}(j) + t_{idle_k} - t_{idle_i}}{N_{WW} + 1} & \text{if } TSR(i) = 1 \end{cases}, \quad (1)$$

where  $n_0(i)$  is the number of bits 0 in the *TSR*,  $t_{ref}$  is the system clock factor,  $i$  and  $k$  indexes stand for the two last moments when the receiving node received data from the sender,  $t_{idle_k}$  and  $t_{idle_i}$  are the sender idle listening time for the two last times it received a *WB* and  $N_{WW}$  is the number of wake-ups without receiving *WB*. The three values  $t_{idle_k}$ ,  $t_{idle_i}$  and  $N_{WW}$  are calculated at the sender side and sent to the receiver in the data frame, whose structure is described in the next section.

The variable  $N_{WW}$  stores the statistic of the received *WB* in order to deal with the *TSR* wrong convergence problem described in Fig. 2. Each time the sender wakes up without receiving the *WB*, the variable  $N_{WW}$  is increased. This information is sent to the receiver at the next data frame and then is reset to 0.

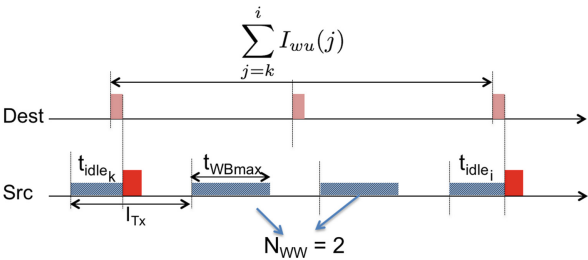


Fig. 4. Adaptive function to calculate  $I_{WU}$ .

In FTA-MAC, the algorithm is designed to converge toward a steady state value of  $I_{WU}$ , where the  $TSR$  contains a sequence  $[11111111 \dots]$ . This sequence is the best sequence of the  $TSR$ , *i.e.* each time the receiver wakes up, it receives the data from the sender. This sequence also specifies that the  $I_{WU}$  has converged. As discussed in Sect. 3.1, the sequence  $[11111111 \dots]$  has the latent problem that the receiver can not detect the increase of traffic rate. But in FTA-MAC protocol, the adaptive function is provided the essential information ( $t_{idle}$ ,  $N_{WW}$ ) of the sender so the receiver can detect as soon as the traffic rate changes (either increases or decreases) and the receiver does not need to wake up more frequently than the sender.

3.3 Frame Structure

The frame structure proposed for FTA-MAC is described in Fig. 5. The first byte is used for the frame information in which the 3 first bits specify the type of frame. In FTA-MAC, there are 3 types of frame: *WB* (000), *DATA* (001) and *ACK* (010). The next 5 bits are reserved for further utilization. Based on the type of frame, the 4 or 8 next bytes are used for address information. The *WB* frame is broadcast so the destination address does not need to be defined. In this case, the frame size of *WB* is 7 bytes and 11 bytes for *ACK* because these frames have not data.

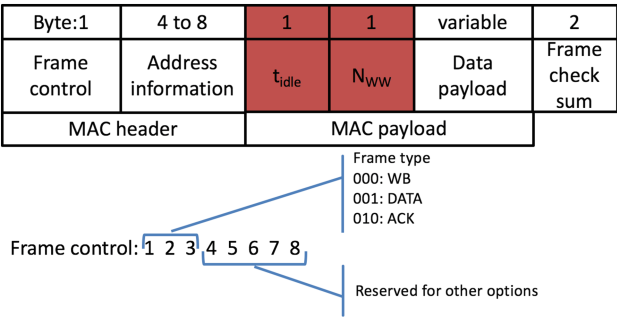


Fig. 5. New frame structure of FTA-MAC.

As discussed in Sect. 3.2 above, the sender will include the 2 variables  $t_{idle}$  and  $N_{WW}$  in the data frame. The 2 first bytes in the payload part are reserved for these variables so the size of the *DATA* frame is greater than 13 bytes. For the simulations of the next section, the size of *DATA* frame is set to 16 bytes.

## 4 Performance Evaluation

In this section, the evaluation of performance of FTA-MAC protocol is presented. This protocol is implemented in the OMNeT++ network simulator using the framework MiXiM for WSNs. For comparison purpose, TAD-MAC and RICER protocols have also been implemented. This section addresses first point-to-point communications, then the same simulation framework is used to validate the protocol for multi-sender communications.

### 4.1 Simulation Topology

OMNeT++ is an object-oriented modular event network simulator and it provides the infrastructure for writing simulations with the component architecture. MiXiM is a modeling framework for wireless network (WSNs, body area network, ad-hoc network, vehicular network, etc.).

Each sensor node is simulated with a single omni-directional antenna and using the simple path loss model of radio propagation. Table 1 summarizes the key parameters which are used to simulate the radio. Most of these parameters are from the data sheet of CC2420 radio chip [13], which is used in popular WSN testbed systems such as MICAz [14], TelosB [15] and PowWow [16]. These parameters are used to calculate the energy consumption in the Physical layer of MiXiM.

**Table 1.** Configuration of radio parameters.

Bitrate	250 kbit/s
RX current	18.8 mA
TX current	17.4 mA
Sleep current	0.03 mA
$P_{Tx}$	1mW

### 4.2 Convergence Speed

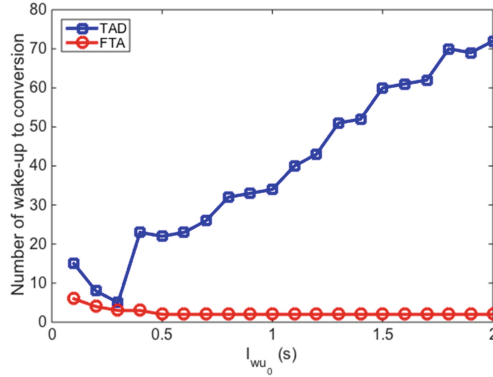
In a network with variable traffics, the convergence speed is an important factor to evaluate the performance of a MAC protocol. The convergence speed corresponds to the number of wake-ups of the receiver needed to converge toward the good *TSR* register, *i.e.* [1111...] for FTA-MAC and [1010...] for TAD-MAC.

**Table 2.** Configuration of MAC protocol parameters.

$TSR$ length	4
$t_{WBmax}$	500 ms
$t_{CCA}$	.5 ms
$I_{WU_0}$	0.1s - 2s

When the receiver reaches the steady state, the  $t_{idle}$  of the sender is minimized such that the faster the receiver converges toward the steady state, the less energy the sender consumes. In order to evaluate the convergence speed of FTA-MAC and TAD-MAC, the first simulation scenario is configured with a fixed traffic rate of 2 frames per second for a point-to-point communication. Table 2 summarizes the MAC protocol parameters which are used in the simulations. These parameters are the values used for TAD-MAC protocol in [11]. The variable  $I_{WU_0}$  is tuned from 0.1s to 2s to study the impact of the initial wake-up interval on the convergence speed of these protocols.

The simulation results in Fig. 6 show that FTA-MAC protocol reaches the steady state value faster than TAD-MAC and the convergence speed is nearly independent of the variable  $I_{WU_0}$ . The convergence speed is constant when the value of  $I_{WU_0}$  is greater or equal than the traffic rate ( $I_{Tx}=0.5$ s for a data rate of 2 frames/s). On the contrary, the convergence speed of TAD-MAC depends on the value of  $I_{WU_0}$ . Even with the best value of  $I_{WU_0}$  (0.3s) for TAD-MAC, the latter converges slower than FTA-MAC.

**Fig. 6.**  $I_{WU}$  convergence speed as a function of  $I_{WU_0}$ .

### 4.3 Energy Evaluation for Variable Traffics

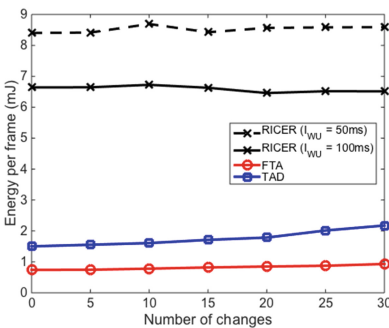
One of the advantages of FTA-MAC compared with TAD-MAC is the fast convergence speed toward a steady state value of  $I_{WU}$ . This advantage can be tra-

**Table 3.** Simulation parameters for variable traffic.

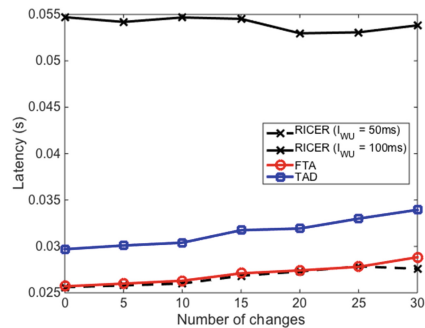
Simulation time	2000 s
Traffic rate	1 frame/s - 10 frames/s
$I_{WU_0}$ (FTA &TAD)	0.3 s
$I_{WU}$ (RICER)	0.05s / 0.1s
Number of changes	from 0 to 30
Number of random simulations	100

duced in a significant energy gain for sending and receiving a frame in a network with variable traffics. The traffic rate changes by increasing or decreasing the  $I_{Tx}$  of the traffic generator of the sender. In this simulation, a change corresponds to a new random value for  $I_{Tx}$  uniformly distributed between 100 ms and 1000 ms. The number of changes is varied from 0 to 30 for a simulation time of 2000 s. All of these parameters are specified in Table 3. The results for variable traffic are shown in Figs. 7 and 8, where each average value is calculated from the results of 100 random simulations. FTA protocol is compared with TAD-MAC and also with the commonly used RICER protocol. RICER has no adaptation mechanism and is tuned by its wake-up interval  $I_{WU}$ .

Figure 7 shows the average energy consumption for sending and receiving a frame which is calculated at both sender and receiver sides for a successful transmission. The energy consumption of TAD-MAC is 1.6 to 2 times more than the one of FTA-MAC and this gain increases with the number of changes of  $I_{Tx}$ . The FTA-MAC protocol saves more energy than TAD-MAC because of the fast convergence speed. For each change of the traffic rate, the receiver needs to adapt its  $I_{WU}$  according to the new steady state value of  $I_{Tx}$ . The fast convergence speed reduces not only the energy consumption of the receiver but also of the sender by reducing the energy wasted during the *idle listening* period.



**Fig. 7.** Energy consumption per frame under variable traffic.



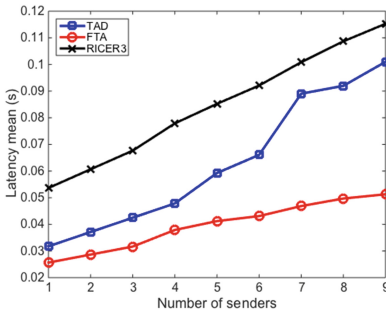
**Fig. 8.** Average latency of frame under variable traffic.

The adaptation of the receiver toward the data rate of the sender allows FTA-MAC to outperform also RICER protocol not only under the variable traffic but also with fixed traffic. Indeed, Fig. 7 shows that FTA-MAC uses nearly 7 times less energy consumption than RICER ( $I_{WU} = 100$  ms) and 9 times less than RICER ( $I_{WU} = 50$  ms).

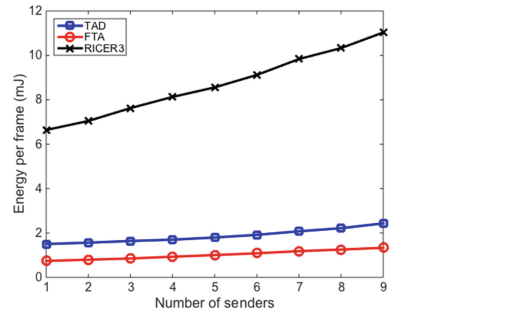
The other factor used to evaluate the performance of the MAC protocols is the frame latency. The results are shown in Fig. 8. Although the RICER sensor node wakes up more frequently, FTA-MAC protocol has a lower latency. To gain 0.002 s less latency than FTA-MAC at 30 changes of traffic rate, RICER sensor node with a  $I_{WU} = 50$  ms needs to wake up 10 times more frequently and thus consumes much more energy (about 9 times more).

#### 4.4 Evaluation for Multi Sender Networks

When the communication is extended to multiple concurrent senders, data collisions can occur and decrease the performance of a protocol. Thus this kind of networks is also addressed to evaluate the performance of FTA-MAC. A star topology with a number of senders increasing from 1 to 9 is used and a traffic rate of 2 frames per second is set for all senders. When increasing the number of nodes, wake-up collisions will appear more frequently. Simulation results are given in Figs. 9 and 10, showing the average latency and the energy consumption of FTA-MAC, TAD-MAC and RICER3 (a version of RICER with a mechanism to avoid data collision). The latency and the energy consumption of FTA-MAC slightly increase while these factors for TAD-MAC and RICER3 sharply increase with the number of senders. In RICER3, the number of slots are set according to the number of senders to avoid data collision. Thus, when many senders concurrently send data to the receiver, some nodes will take a long time to wait their  $WB$  (*i.e.* the  $t_{idle}$  is increased). For TAD-MAC protocol, the data collision increases the time to converge toward the steady state. This issue increases the latency and the energy consumption in TAD-MAC.



**Fig. 9.** Average latency of frame with multiple concurrent senders.



**Fig. 10.** Energy consumption per frame with multiple concurrent senders.

## 5 Conclusion

In this research, a novel adaptive energy efficient MAC protocol is introduced. The FTA-MAC protocol presents an algorithm that allows the receiver to adapt its  $I_{WU}$  according to the transmission rates of the sender. With the fast convergence speed, the receiver is well scheduled to wake up closely to the sender so reducing its idle listening. Thus the energy consumption in a sensor node is decreased 2 times compared to TAD-MAC protocol and 9 times compared to RICER protocol. This improvement is even more interesting for communications with variable traffics and also for a star network topology with concurrent multi-senders. The further work will extend FTA protocol to multi-hop networks, where the impact of idle listening on the energy consumption increases with the number of hops.

## References

1. Definition of the Internet of Things (IoT). <http://iot.ieee.org/definition.html>. Accessed on 29 Nov 2015
2. Bachir, A., Dohler, M., Watteyne, T., Leung, K.K.: MAC essentials for wireless sensor networks. *IEEE Commun. Surv. Tutorials* **12**, 222–248 (2010)
3. Ye, W., Heidemann, J., Estrin, D.: An energy-efficient MAC protocol for wireless sensor networks. In: *Proceedings of the 21st IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1567–1576, June 2002
4. van Dam, T., Langendoen, K.: An Adaptive energy-efficient MAC protocol for wireless sensor networks. In: *Proceedings of the 1st ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 171–180, November 2003
5. Polastre, J., Hill, J., Culler, D.: Versatile low power media access for wireless sensor networks. In: *Proceedings of the 2nd ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 95–107, November 2004
6. Buettner, M., Yee, G.V., Anderson, E., Han, R.: X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In: *Proceedings of the 4th ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 307–320, October 2006
7. El-Hoiydi, A., Decotignie, J.-D.: WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks. In: *Proceedings of the 9th International Symposium on Computers and Communications (ISCC)*, pp. 244–251, June 2004
8. Tang, L., Sun, Y., Gurewitz, O., Johnson, D.B.: PW-MAC: an energy-efficient predictive-wakeup MAC protocol for wireless sensor networks. In: *Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1305–1313, April 2011
9. Lin, E.-Y. A., Rabaey, J.M.: Power-efficient rendezvous schemes for dense wireless sensor networks. In: *Proceedings of the 11th IEEE International Conference on Communications (ICC)*, pp. 3769–3776, June 2004
10. Sun, Y., Gurewitz, O., Johnson, D.B.: RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In: *Proceedings of the 6th ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 1–14, November 2008

11. Alam, M.M., Berder, O., Menard, D., Sentieys, O.: TAD-MAC: traffic-aware dynamic MAC protocol for wireless body area sensor networks. *IEEE J. Emer. Sel. Topics Circuits Syst.* **2**(1), 109–119 (2012)
12. E.-Y. Lin, J. Rabaey, S. Wiethoelter, Wolisz, A.: Receiver initiated rendezvous schemes for sensor networks. In: *Proceedings of the 48th IEEE Global Telecommunications Conference (GLOBECOM)*, vol. 5, pp. 3122–3128, December 2005
13. CC2420 Single-Chip 2.4GHz. <http://www.ti.com/product/CC2420/technicaldocuments>. Accessed on 29 Nov 2015
14. MEMSIC, Inc - WSN Nodes — MPR2400CB. <http://www.memsic.com/wireless-sensor-networks/MPR2400CB>. Accessed on 29 Nov 2015
15. MEMSIC, Inc - WSN Nodes — TPR2420. <http://www.memsic.com/wireless-sensor-networks/TPR2420>. Accessed on 29 Nov 2015
16. Power Optimized Hardware, Software FrameWork for Wireless Motes. <http://powwow.gforge.inria.fr/>. Accessed on 29 Nov 2015