Duo-MAC: Energy and Time Constrained Data Delivery MAC Protocol in Wireless Sensor Networks

Messaoud Doudou[ab], Mohammad Alaei[c], Djamel Djenouri[b], Jose M. Barcelo-Ordinas[c], Nadjib Badache[ab]
a: University of Sciences and Technology Houari-Boumediene (USTHB), Algiers, Algeria
b: CERIST Research Center, Algiers, Algeria
c: DAC Department, Universitat Politcnica de Catalunya (UPC), Barcelona, Spain
E-mails: {doudou,badache}@cerist.dz, ddjenouri@acm.org, {malaei,joseb}@ac.upc.edu

***Abstract*—We present Duo-MAC, an asynchronous cascading wake-up scheduled MAC protocol for heterogeneous traffic forwarding in low-power wireless networks. Duo-MAC deals with energy-delay minimization problem and copes with transmission latency encountered by Today's duty-cycled protocols when forwarding heterogeneous traffic types. It switches, according to the energy and delay requirements, between Low Duty cycle (LDC) and High Duty Cycle (HDC) operating modes, and it quietly adjusts the wake-up schedule of a node according to (i) its parent's wake-up time and (ii) its estimated load, using an effective real-time signal processing linear traffic estimator. As a second contribution, Duo-MAC, proposes a service differentiation through an improved contention window adaptation algorithm to meet delay requirements of heterogeneous traffic classes. Duo-MAC's efficiency stems from balancing between the two traffic award operation modes. Implementation and experimentation of Duo-MAC on a MicaZ mote platform reveals that the protocol outperforms other state-of-the-art MAC protocols from the energy-delay minimization perspective.**

***Index Terms*—Wireless Sensor Networks; MAC; Asynchronous Protocols; Real Time; QoS; Delay;**

I. Introduction

Current wireless sensor networks (WSN) applications, ranging from health, to industrial monitoring, transportation, and automation, [1], are complex and have several requirements. In addition to energy constraint that was almost the only metric for consideration in earlier applications, new constraints appear with the emerging applications, such as latency, security and reliability. In these applications, there is a real need to support both periodic and event-driven data forwarding, and nodes tend to carry multiple types of traffic such as real-time RT (e.g. detected events) and non real-time NRT (e.g. periodic monitoring reports). However, the co-existence of such traffic patterns makes it difficult to meet the requirements in terms of energy and delay. Although service differentiation enables each type of traffic to be sent according to its degree of importance, this gain in delay is often lost because the sender spends more time waiting to its receiver wake-up.

Given that the radio interface is likely to consume the largest amount of a node's battery [2], most of WSN MAC protocols use duty-cycling to save energy. Channel scheduling has to deal with duty-cycling that has a direct impact on the forwarding delay of a node. The forwarding delay problem is targeted herein, where a novel two-state asynchronous cascading wake-up scheduled MAC protocol, called Duo-MAC, is presented. The proposed protocol also considers energy-delay minimization. In absence of real-time traffic, Duo-MAC runs in a low duty-cycling mode and behaves as an energy-efficient MAC but still low-latency following an EEF (Energy Efficiency First) strategy. Whereas when an event is detected, a node enters in a high duty-cycling mode and behaves as a delay-efficient MAC to forward real-time traffic following an DEF (Delay Efficiency First) strategy. In addition to operating in two modes, Duo-MAC ensures quick packet forwarding by adopting a dynamic cascading of active periods and quietly adjusts the wake-up time of a node according to (i) its parent's wake-up time and (ii) its estimated load, using an effective real-time signal processing linear traffic estimator. Finally, to meet the dynamic traffic requirements of each class, Duo-MAC processes each packet according to its degree of importance by using an effective service differentiation based on hybrid prioritization scheme, and it adapts the contention window size of each traffic class based on an effective link quality estimator. This captures the real-time link characteristics, reducing thus collisions as well as the latency.

The remainder of the paper is organized as follows. Section II presents related work. Section III introduces the proposed protocol, and Section IV presents its implementation on MicaZ mote platform, as well as the experimental results. Finally, Section V draws the conclusions.

II. Related Works

Depending on how a sender joins its intended receiver, duty-cycled contention-based MAC protocols can be classified into synchronous vs. asynchronous protocols. Synchronous contention-based MAC protocols specify and coordinate active/sleep periods by exchanging SYNC packets for synchronization. These protocols cause high contentions and collisions by grouping communications in short periods. On the other hand, asynchronous low-power protocols have no overhead

978-1-4673-2480-9/13/$31.00

for synchronization, which explains their wide utilization and implementation in many WSN operating systems, e.g., TinyOS LPL (Low power Listening) [3]. While the synchronous protocols are heavily impacted by clock drift [4], asynchronous protocols experience no such effects. Nonetheless, in asynchronous protocols, communicating nodes are totally decoupled, which may significantly increase the delay for the sender to meet the receiver's active period. The delay analysis of asynchronous MAC protocols has been investigated in our previews study [4] where asynchronous protocols have been classified according to the delay-efficiency-related perspective. We reported six different categories of asynchronous MAC protocols; static wake-up preamble, e.g., B-MAC [11], adaptive wake-up preamble, such as WiseMAC [13], collaborative schedule setting, e.g., LL-MAC [5], collision resolution like Sift [14], receiver-initiated such as A-MAC [15], and anticipation-based protocols [16].

It has been shown that collaborative schedule setting protocols further improve the delay over other classes and ensure end-to-end delay decrease. Collaborative schedule setting means that neighbor nodes collaborate to set their wake-up schedules in a cascading way. However, none of these protocols considers traffic load of intermediate nodes when targeting the delay, and few of them considers the wake-up time minimization (e.g. LL-MAC). Moreover, QoS differentiation has not been integrated within these protocols. The literature presents many QoS-aware MAC protocols for WSNs, e.g., [9]. We refer to [6] for a review on QoS-aware MAC. There are several mechanisms, [12], essential for providing QoS in terms of minimum delay at MAC layer when multi-traffic classes are present. First of all, the nodes need to minimize the wake-up waiting time between a transmitter and a receiver. Second, the node has to perform intra-traffic class differentiation by means of intra-queue management in order to provide traffic priority at the node level. Finally, the node has to perform inter-node traffic class differentiation. The aim of Duo-MAC is firstly to minimize the waiting time at MAC layer by making the transmitter and receivers wake-up times as closer as possible using an effective and dynamic cascading wake-up schedule. Secondly, inter-node traffic differentiation is achieved by adaptively adjusting the contention window (CW) according to the link characteristics and the different traffic classes delay requirements. And finally, a intra-queue priority queuing is used in order to differentiate the different classes of traffic.

III. Duo-MAC Protocol

Duo-MAC is a contention-based MAC protocol which adapts its wake-up schedule to network traffic conditions. In front of heterogeneous NRT and RT traffics, the protocol trades off energy-efficiency with latency. MAC protocols like B-MAC [11] are energy-efficient in the sense that nodes stay in the sleep mode as much as possible and periodically poll the channel for eventual incoming data packets. This polling period determines the wake-up latency that a packet has.

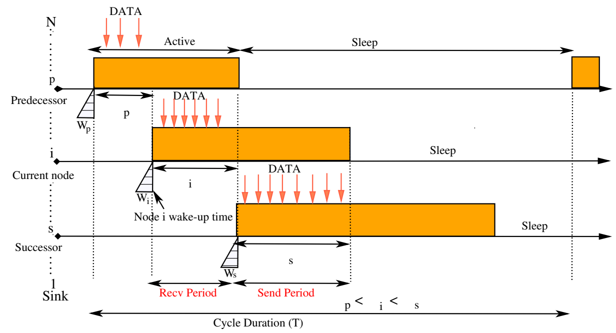

Fig. 1. Cascading Wake-up in Duo-MAC

Scanning the channel more frequently reduces the latency at the cost of energy.

Duo-MAC switches between two operating modes, i) a Low Duty Cycle (LDC) state whose aim is energy-efficiency for NRT traffic and, ii) a High Duty Cycle (HDC) state to optimize network performance in terms of energy saving and delay guarantee. Initially, a node is in the initialization state when it is powered-on. After the set-up state, a node enters the LDC state and switches periodically between active and sleep operation. This state is designed for NRT traffic and for best-effort traffic, for which Energy-Efficiency First (EEF) strategy is applied. On the other hand, whenever a node detects RT packets, the node enters the HDC state and applies Delay-Efficiency First (DEF) strategy, where it increases its wake-up rate and extends its duty-cycle period to meet RT traffic requirements at the cost of more power-consumption. The end of real-time traffic is signaled by a source node and propagated to the forwarders, causing these intermediate nodes to return back to LDC state.

The dynamically updated cascading wake-up feature of Duo-MAC wakes-up nodes asynchronously and independently from their neighbors to receive and send data packets. A node broadcasts its next scheduled time in the vicinity so that predecessors and new joined nodes can update their schedules. Each operation cycle $\mathbf{T}$ is divided into an active and a sleep period of durations T_{Active} and T_{Sleep} with a duty-cycle ratio of $\frac{T_{Active}}{T}$. The active period is divided into two intervals, T_{Recv} and T_{Send}, (see Fig.1). Nodes content to send NRT packets using CSMA, where each transmission includes DATA packet and its acknowledgment ACK. Note that the length of these intervals is not fixed but varies according to the mode of operation and to the traffic load, which makes the duty-cycle in our protocol adaptive and energy-delay balancing.

A. Cascading Wake-up Schedule

To eliminate the delay wasted when a transmitter remains waiting for the receiver to wake-up, the active time of both nodes must be close to the propagation time value, plus the time to receive a packet. Duo-MAC is based on adaptive self-adjusting cascading wake-up scheduling, inspired from a control loop system using an effective signal processing linear traffic load estimator. When a new node joins the network it enters the initialization phase and builds the neighbor table storing neighbors' scheduling information. Existing nodes may be in the initialization phase or in the periodic active and sleep phase. A node monitors the channel and receives beacon packets from its neighborhood. The beacon packet of a node i contains its *ID*, its hop-count h_i defined as the number of hops from node i to the sink, and its wake-up offset w_i (the remaining time until its next wake-up time). Minimum hop-count routing protocol is chosen as a basis protocol. Choosing minimum hop-count is reasonable and is justified by the objective in minimizing the end-to-end delay.

Let the source node be labeled N and the sink labeled 1. Then, let **S** be the list of successors nodes (i.e. forwarders), S={s∈ i's neighbors table, $h_s < h_i$}, and **P** a list of predecessors, P={p∈ i's neighbors table, $h_p > h_i$}. If s is the selected next forwarder node of node i, then the wake-up time w_i must be shifted σ times before w_s. In Duo-MAC, σ is the necessary time to receive non-realtime (NRT) traffic from all predecessors (Fig.1). Consequently, the more the node gets closer to the sink, the more its corresponding σ time will increase, as the relay node attempts to forward more traffic compared to other nodes. The wake-up time w_i of a node is not static, but it depends on the traffic load to be handled at the current cycle and the wake-up time of its successor, which make the wake-up schedule adaptive to the traffic to be handled at each node. As a result, w_i can be calculated by the following function: $w_i = f(\sigma_i, w_s)$. Note that $\sigma_i(t_n)$ is related to the amount of traffic load, $L_i(t_n)$, to be received by a node from all predecessors at a given operation cycle t_n, which may vary in time. For instance, a node needs to estimate the upcoming traffic load for the next operation cycle so that it can calculate its next wake-up time. Later, in subsection III-E, a description of how a node can estimate its upcoming traffic load will be given.

B. Low Duty Cycle Mode

In LDC mode, every cycle contains an active period, a sleep period, and some periodic quick channel polling. While the active period is basically used to send NRT traffic, the periodic channel polling is used to check for possible incoming RT traffic in the middle of the sleep period. The number of channel polling slots must tradeoff energy with delay. A node wakes-up exactly at its scheduled time (updated at each cycle) and sends a *RECV* beacon indicating the beginning of T_{Recv} period. A node then is ready to receive data from its predecessor nodes. At the end of T_{Recv} period, a node calculates according to the node's NRT time traffic its next wake-up time schedule and broadcasts it using *SEND* beacon. The *RECV* beacon signals to predecessors a current wake-up time of a node, as depicted in Fig.2, while the *SEND* beacon's role is twofold. First, it indicates the end of T_{Recv} period and the beginning of T_{Send} period. Second, it contains information about the next wake-up time, as well as hop-count of a node, and some additional information. Note that the two beacons are important to calculate Δt, which is the time difference between the current wake-up time of a successor and the end of T_{Recv} period. Δt is

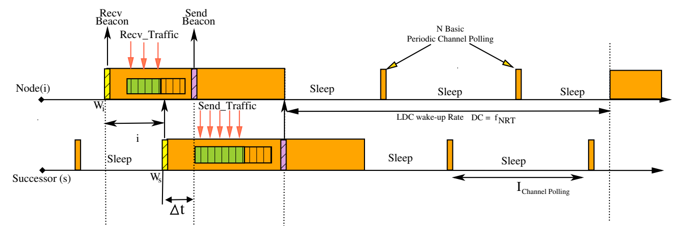

Fig. 2. Low Duty Cycle Mode

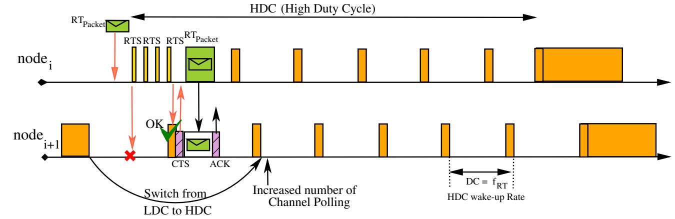

Fig. 3. High Duty Cycle Mode

a delay a node spends in idle listening waiting for its successor or predecessor to be ready to communicate. This time must be minimized as much as possible and can be used to adjust a node wake-up time. An event can occur at any unpredicted instant, causing the generation of RT traffic. If a node is in its active time, it switches to HDC mode and RT traffic is forwarded immediately, taking advantage of the overlapping of the cascading wake-up. If a node is in its sleep period, its successor would also be sleeping and RT traffic must wait until the next active period. As a consequence, a first RT packet of the event may suffer some delay and it may not meet its deadline. To cope with this, we propose to use periodic quick channel polling in the middle of the sleep period to detect possible incoming real-time packets, as shown in Fig.2. For instance, each node performs N channel polling before its next wake-up time. Consequently, the channel polling of nodes in the path are automatically set in cascading way and a worst-case delay can be reduced from, T_{Sleep}, to $I_{Channel\_Polling}$, the interval between two successive channel polling, where $I_{Channel\_Polling} = \frac{T_{Sleep}}{N}$. Note that the bigger the value of N is, the smaller the worst-case delay will be but at the cost of more energy consuming and inversely.

C. High Duty Cycle Mode

A node can detect an event and generates an RT packet, or it can receive an RT packet during its periodic channel polling. In both cases, a node switches to HDC mode, then it uses a succession of RTS packets as a wake-up preamble to reach its forwarder node. If a node during one of its channel polling receives such an RTS packet, it responds with CTS. A channel polling period must be at least equal to the transmission time of RTS, plus the interval time between two RTS packets. This is to ensure the wake-up packet to be heard (Fig.3). The data frame in Duo-MAC contains three fields: *Traffic_type*, *Traversed_Hop_Number*, and *More_bit*. Once the RT packet is received, a node goes immediately to HDC mode and sets its wake-up rate to f_{RT}, following the DEF strategy. If a node is in its LDC active period, then it can immediately forward RT packets, using service differentiation with priorities (as described later in Section III-F). However, if a node is in the sleep period and receives RT packets during its periodic channel polling, it must use RTS wake-up strobes to join its next hop. In this case, the receiver replies with a CTS and keeps the radio on. Then the sender transmits the data packet upon receiving the CTS and waits for the data acknowledgment, ACK. A sender indicates each time it sends an RT packet that there is more RT packets to be sent using the flag *More_bit*. Once this field is set to 0, each node receiving this packet knows that it is the last RT packet and hence switches back to LDC mode.

D. Estimating the Traffic Load

To determine the best traffic load estimator, we have performed extensive experimentations using our $20-MicaZ$ Testbed, where the traffic load flowed by the nodes has been tracked. In this experiment, a tree topology is used and each node except the sink is generating traffic with a frequency of $2pkt/sec$. The average loads in some nodes are depicted in Fig. 4. From this figure, it can be noticed that the traffic load has a linear form. For this purpose, we have decided to choose a linear estimator because it enables us to adequately estimate traffic in our scenario. However, other estimators that fit other kinds of traffic estimation can be easily integrated in our protocol. Each node in the network can estimate its incoming traffic. At a given operation cycle t_n, the node gathers information about the received traffic, $L_i(t_n)$ at node i. To calculate the estimated traffic for the next cycle t_{n+1}, we propose to predict the current load value of the nodes according to M previous load values using a linear prediction scheme. Assume that the load values $L_i(t_n)$, at times t_n, with n=1,..,M, (respectively) are known, and the load value of the node at time t_{n+1} has to be predicted. The load $L_i(t_{n+1})$ of node i at time t_{n+1} is calculated using the least mean square (LMS) method: $L_i(t_{n+1}) = at_{n+1} + b$, where coefficients a and b are obtained from solving the system equations of eq.(1):

$$\left\{ \begin{array}{lll} \sum_{n=1}^{M} t_n L_i(t_n) & = & a\sum_{n=1}^{M} t_n^2 + b\sum_{n=1}^{M} t_n \\ \sum_{n=1}^{M} L_i(t_n) & = & a\sum_{n=1}^{M} t_n + bM \end{array} \right. \quad (1)$$

In the experiment, M has been fixed to 100, a and b have been accordingly calculated.

E. Adjusting The Wake-up Time

As discussed in section III-A, the wake-up time w_i of node i can be calculated as a function of two parameters, $w_i = f(\sigma_i, w_s)$, where $\sigma_i(t_n)$ is the required time to receive the estimated load $L_i(t_{n+1})$ traffic for node i for the next cycle t_{n+1}. Suppose TX_i is the required time to transmit one packet from an arbitrary node i including contention t_{cw}, t_{rts}, t_{cts}, t_{data}, t_{ack}, and 3*IFS (Inter Frame Space). Then, σ can be calculated as follows,

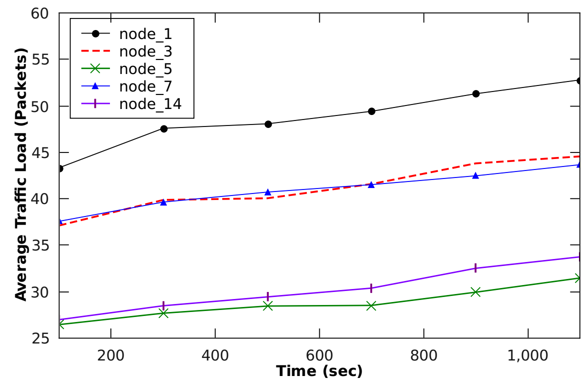

Fig. 4. Average Load at each node

$$\sigma_i(t_{n+1}) = T_{Recv} + TX * L_i(t_{n+1}) + T_{Send} \quad (2)$$

Where T_{Recv} and T_{Send} are the required times to transmit the receive beacon and the send beacon respectively, including contention times. Given the value of $\sigma_i(t_{n+1})$, the next wake-up time, $w_i(t_{n+1})$, can be calculated as:

$$\begin{aligned} w_i(t_{n+1}) =& f\left(\sigma_i(t_{n+1}), w_s(t_{n+1})\right) \\ =& w_s(t_{n+1}) - \sigma_i(t_{n+1}) + \Delta t_n \end{aligned} \quad (3)$$

Δt is the delay a node spends in idle listening waiting for its successor or predecessor to be ready to communicate. It can also be viewed as the a time between sending its T_{Send} beacon and receiving the T_{Recv} beacon of its forwarder. If a node receives T_{Recv} before sending the T_{Send}, then the value of Δt is negative. This means that a node must decrease its wake-up time by Δt. Whereas, if a node sends the T_{Send} before receiving T_{Recv}, then the value of Δt is positive, i.e., the node must increase its wake-up time by Δt.

F. Service Differentiation and CW Adaptation

In order to provide hybrid prioritization levels, per traffic type and per traversed hop-count, an effective service differentiation scheme that relies on contention window (CW) size adaptation mechanism is proposed. Saxena et al., [8], propose a CW adaptation algorithm based on loss probabilities by defining for each traffic class, a CW_{target} to be reached in many steps. Yigitel et al., [9], propose a CW adaptation approach that initiates the CW of each traffic class from the medium value, $\frac{cw_{min}+cw_{max}}{2}$, then it moves it up and down according to PRR (Packet Reception Ratio) measurements. We propose a CW adaptation algorithm similar to that of the literature. However, the proposed algorithm defers from those in [8], [9] in two main points; first, instead of using collision rate or PRR, we propose to combine the PRR with a novel link quality metric called link burst length, b_{min} and b_{max} defined in [10]. With this metric, a good link is characterized with a short b_{max}, which defines the maximum consecutive packet transmission errors in a burst, and with a long b_{min} which defines the minimum consecutive successful packet transmissions between two bursts of packets. For example, for a burst of 10 transmitted packets, [10], the sequence 0110010011 represents with a 0 at position i-th that packet i-th has failed and a 1 at position i-th that packet i-th has correctly arrived. In this example, b_{max}=2 and b_{min}=1 for packet bursts of window size equal to three. Paper [10] shows that these two parameters outperform PRR and capture better the link quality. Second, the CW is initiated to cw_{min}, to go up and down based on the estimated link quality.

$$P_{LQ}(t_i) = PRR * \frac{b_{min}}{b_{max}} \tag{4}$$

Summarizing, Duo-MAC runs the CW adaptation Algorithm 1 each N traffic samples, which constitutes a burst of length N. For each burst the algorithm computes PRR, b_{min}, and b_{max}, and uses the link quality probability P_{LQ} given by Eq.(4) as a quality metric to accordingly move the CW.

Finally, Duo-MAC uses the same principle as introduced in [9] when setting non-overlapping CW sizes, where $CW(RT) < CW(NRT)$. The adaptation coefficient are chosen while fulfilling the following conditions. $\alpha_{up}(RT) < \alpha_{up}(NRT)$ and $\alpha_{down}(RT) > \alpha_{down}(NRT)$.

IV. Performance Evaluation

A. Setup and Parameters

In order to evaluate the performance of Duo-MAC protocol under different scenarios and network conditions, we have implemented the Duo-MAC protocol and have compared it with LL-MAC [5] – the most similar solution to Duo-MAC from the literature – as well as an energy-unconstrained CSMA-like protocol defined in, [3], TinyOS-2.x. The evaluation is carried out with both simulations and experimentations on MicaZ motes. First, Avrora has been used [7]. It accurately emulates the AVR cycle execution and the RF CC2420 physical layer [2]. Table I sketches the simulation setup. CSMA serves as the bound of delay efficiency, given that it causes no waiting delay due to its full-duty-cycling. In the simulation, different configurations are evaluated using Avrora. Evaluation has also been performed using a Testbed in which nodes are placed on top of walls in several offices in our campus. We remark that communication links can have relatively high loss rates (up to 20%). In the experience, 50% of the nodes are source of CBR traffic, 25% is NRT and the rest is RT traffic (events). A node generates for each event 3 to 5 packets. Each experiment is repeated multiple times and each point in the following curves comes from the average result of all experiments.

B. Results and Discussions

Configuration A: For this configuration, traffic generation of RT and NRT are fixed to $1/15$ Pckts/s and $1/5$ Pckts/s, respectively. The number of nodes is varied from 20 to 100, resulting in a dense network topology of up to 15 hops. Fig. 5 shows the average latency measurement vs. the number of nodes. It can be observed that the latency provided by Duo-MAC for RT traffic is 50% lower than that of LL-MAC, and 5% over the unconstrained-CSMA, and that the latency

Algorithm 1: CW Adaptation Algorithm

Init: $cw_{cur} = c_{min}$
1: Calculate $P_{LQ}(t_i)$
2: if $(P_{LQ}(t_i) < P_{LQ}(t_{i-1}))$
  link quality is good so move cw down
3:   $\Delta cw = \alpha_{down} * (cw_{min} - cw_{cur})$
 else
  link quality is poor so move cw up
4:   $\Delta cw = \alpha_{up} * (cw_{max} - cw_{cur})$
 end if
5: $cw_{cur} = cw_{cur} + \Delta cw$

TABLE I
Simulation parameters.

Parameter	Value
Monitoring area	$150 \times 150m^2$
Deployment type	Grid
Network size	20 to 100 Nodes
Packet size	28 Bytes
Routing algorithm	Min Hop Count
Max. communication range	15m
Queue weights (RT/NRT)	0.7 / 0.3
CW_{min} (RT/NRT)	4 / 12
CW_{max} (RT/NRT)	12 / 24
α_{up} (RT/NRT)	0.3 / 0.7
α_{down} (RT/NRT)	0.7 / 0.3
Burst length N	16 Packets
a / b coefficients	0.9710 / 0.3735

for NRT is 30% lower than LL-MAC and 20% higher than CSMA for relatively low packet generation frequency.

Configuration B: Here the configuration A has been repeated, but in real environment under harsh conditions (bad links). The experiments are repeated several times during 48 hours running for each one. Only the delay of Duo-MAC and LL-MAC has been evaluated, and plotted in Fig. 6. The latency of Duo-MAC and LL-MAC for NRT are very close with a little improvement of Duo-MAC, up to 5%. This can be explained by the fact that NRT traffic is grouped into the same period for all nodes in a given level of a collection tree, which may lead to more contentions and interferences. This affects the link quality and rises the contention window (of algorithm 1), resulting in a channel access delay. However, Duo-MAC gives the lowest latency for RT traffic, which is 40% lower than LL-MAC. This is due to periodic channel polling strategy that accelerates packet forwarding whenever an event is triggered.

Configuration C: In this configuration, instead of varying the number of nodes, the packet generation of RT and NRT traffic has been increased from 1 Pckts/s to 100 Pckts/s, while the number of nodes has been fixed to 100. Results are depicted in Fig. 7. Here, the latency of Duo-MAC outperforms CSMA (with Binary Backoff) from 20% to 10% lower for RT traffic. But it slightly exceeds CSMA for NRT, 5% to 10% higher. This can be explained by the fact that CSMA

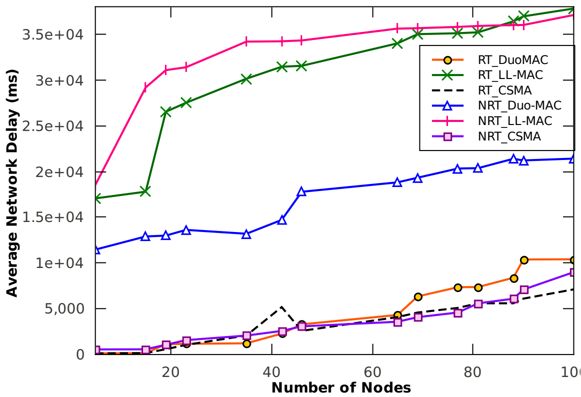

Fig. 5. The average latency vs. increasing number of nodes using Avrora

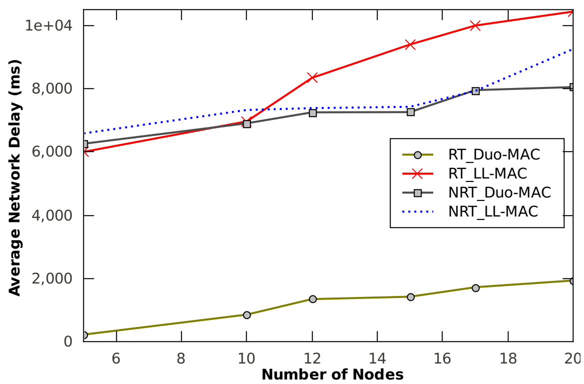

Fig. 6. The average latency vs. number of nodes using TestBed

and Duo-MAC in LDC mode are impacted with contentions. On the other hand, Duo-MAC gives better latency compared to LL-MAC, up to 50% higher for RT and 30% for NRT.

Configuration D: The energy consumption of Duo-MAC, LL-MAC, and CSMA has been evaluated. The average duty-cycle of the network has also been measured and the results are depicted in Fig. 8. The duty-cycle of Duo-MAC for RT traffic and low node number (bellow 40) traffic is higher compared to LL-MAC. This is because all nodes are close to the sink and remain awake to forward events. The duty-cycle quickly decreases with the increase of the number of nodes, since only nodes carrying events remain awake. For NRT traffic, Duo-MAC's duty-cycle goes from 7.31% to 3.06% vs. 5.03% to 2.56% for LL-MAC. This can be explained by the dynamic adjustment of the receive period according to the estimated traffic load in Duo-MAC. Note that the duty-cycle of CSMA is always 100% and has been omitted from the figures for the purpose of not affecting the y axis scale.

V. Conclusion

Duo-MAC, an asynchronous cascading wake-up medium access control (MAC) protocol for wireless sensor networks (WSN), has been proposed in this paper. It targets applications

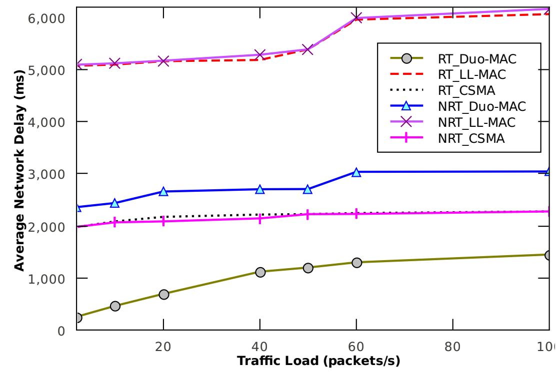

Fig. 7. The average latency vs. traffic load

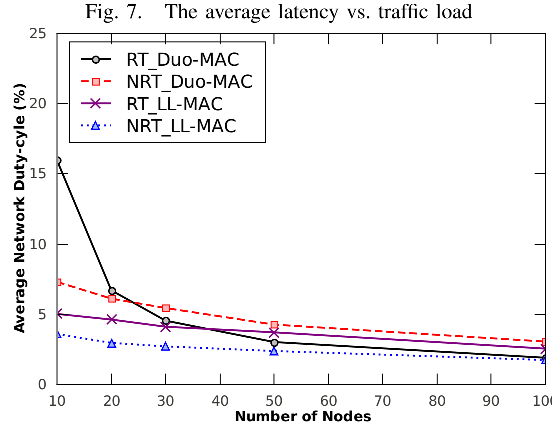

Fig. 8. The average duty-cycle vs. increasing the number of nodes in the network

with heterogeneous traffic by providing traffic differentiation QoS. It achieves energy-time constrained data delivery, and balances energy-efficiency with delay-minimization by adaptively switching between two states according to the dominating traffic in the network. In absence of real-time traffic, Duo-MAC runs in a low duty-cycling mode and behaves as an energy-efficient MAC following the EEF strategy (Energy Efficiency First). As soon as an event occurs, the detecting node enters in a high duty-cycling mode and behaves as a delay-efficient MAC to forward real-time traffic following the DEF strategy (Delay Efficiency First). The nodes waves then the DEF strategy throughout the path towards the sink.

In addition to operating in two modes, Duo-MAC ensures quick packet forwarding by adopting a dynamic cascading of active periods and adjusting the nodes' wake-up time. It also uses an effective service differentiation based on hybrid prioritization scheme, which enables to processes each packet according to its degree of importance. It adapts the contention window size of each traffic class based on an effective link quality estimator. Duo-MAC uses polling during sleep period, which permits to accelerate realtime traffic forwarding. The protocol has been tested and evaluated using simulation and experiment on real sensor motes, where it has been compared with some state-of-the-art protocols; LL-MAC and, a full-duty-cycled CSMA serving a reference for low latency. Results show clear reduction of the delay over LL-MAC for all types traffic, and even some reduction over CSMA for realtime traffic. The gain comes at the cost of a moderate additional energy consumption vs. LL-MAC due to increasing duty-cycling, but the power gain for Duo-MAC over CSMA is much more important.

REFERENCES

[1] V. Gungor, and G. Hancke, "Industrial wireless sensor networks: challenges, design principles, and technical approaches," *IEEE Trans. on Industrial Elect.*, Vol.56, No. 10, pp 4258-4265, 2009.

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

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

[4] M. Doudou, D. Djenouri, and N. Badache, "Survey on Latency Issues of Asynchronous MAC Protocols in Delay-Sensitive Wireless Sensor Networks," *Comm. Surveys & Tutorials*, In Press, 2012.

[5] Q. Yu, C. Tan, H. Zhou, "A Low-Latency MAC Protocol for Wireless Sensor Networks," *WiCom'07*, Shanghai, Sept. 21-25, 2007, pp. 2816-2820.

[6] M.A. Yigitel, O.D. Incel, and C. Ersoy, "QoS-aware MAC protocols for wireless sensor networks: A survey," *Computer Networks*, Vol. 55, No. 8, pp. 1982-2004, 2011.

[7] Avrora, The AVR Simulation and Analysis Framework, URL: "*http://compilers.cs.ucla.edu/avrora/*"

[8] N. Saxena, A. Roy, J. Shin, "Dynamic duty cycle and adaptive contention window based QoS-MAC protocol for wireless multimedia sensor networks," *Computer Networks*, Vol. 52. No. 13, pp. 2532-2542, 2008.

[9] M.A. Yigitel, O.D. Incel and C. Ersoy, "Design and Implementation of a QoS-aware MAC Protocol for Wireless Multimedia Sensor Networks," *Computer Communications*, Vol. 34, pp. 1991-2001, 2011.

[10] S. Munir, S. Lin, E. Hoque, S. Nirjon, J. Stankovic, K. Whitehose, "Addressing Burstiness for Reliable Communication and Latency Bound Generation in Wireless Sensor Networks," *IPSN'10* Sweden, Apr. 2010.

[11] J. Polastre, J. Hill and D. Culler, "Versatile low power media access for wireless sensor networks," *SensSys04*, USA. November 03-05,2004, pp 95-107.

[12] M. Cesana, A. E. C. Redondi, N. Tiglao, A. M. Grilo, J. M. Barcelo-Ordinas, M. Alaei, P. Todorova, "Real-time Multimedia Monitoring in Large-Scale Wireless Multimedia Sensor Networks: Research Challenges," *IEEE NGI12*, Karlskrona, Sweden, 25-27 June, 2012.

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

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

[15] P. Dutta, S. Dawson-Haggerty, Y. Chen, C. Liang, and A. Terzis, "Design and Evaluation of a Versatile and Efficient Receiver-Initiated Link Layer for Low-Power Wireless," *Sensys'10*, Zurich, Switzerland, Nov. 3-5, 2010.

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