iQueue-MAC: A Traffic Adaptive duty-cycled MAC Protocol With Dynamic Slot Allocation

Shuguo Zhuo, Zhi Wang, Ye-Qiong Song, Zhibo Wang, Almeda Luis

iQueue-MAC: A Traffic Adaptive duty-cycled MAC Protocol With Dynamic Slot Allocation

Shuguo Zhuo[†], Zhi Wang[†], Ye-Qiong Song[‡], Zhibo Wang[†] and Luis Almeida[§]
[†]State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou, CHINA
Email: zhuosgzju@gmail.com, {wangzhi, zbwang}@iipc.zju.edu.cn
[‡]LORIA, University of Lorraine, Campus Scientifique, Vandoeuvre, France
Email: song@loria.fr
[§]Instituto de Telecomunica, Universidade do Porto, FEUP, Porto, Portugal
Email: lda@fe.up.pt

***Abstract*—Duty-cycling technique has been widely adopted in MAC protocols for wireless sensor networks to conserve energy. However, low duty-cycle also leads to limited throughput in most of existing solutions. In this paper, we propose iQueue-MAC to provide immediate yet energy-efficient throughput enhancement for dealing with burst or heavy traffic. Combined with CSMA/CA, iQueue-MAC makes use of queue length of each sensor node and allocates suitable TDMA slots to them for packets transmission. During light traffic period, no extra slots will be allocated; iQueue-MAC acts like other low duty-cycle MACs to conserve power. While in burst or heavy traffic period, iQueue-MAC senses the build up of packet queues and dynamically schedules adequate number of slots for packet transmission. We have implemented iQueue-MAC on STM32W108 chips that offer IEEE 802.15.4 standard communication. We set up several real-world experimental scenarios, including a 46 nodes multi-hop test-bed for simulating a general application, and conducted numerous experiments to evaluate iQueue-MAC, in comparison with other traffic adaptive duty-cycle protocols, such as multi-channel version RI-MAC and CoSenS. Results clearly show that iQueue-MAC outperforms multi-channel version of RI-MAC and CoSenS in terms of packet delay and throughput.**

I. Introduction

In wireless sensor networks (WSN), keeping nodes in low duty-cycle, i.e., interleaving very short active and long sleeping periods, is the most efficient way to save energy, thus prolonging network lifetime. However, there is a cost to pay in both low network throughput and extra packet delay, since the network nominal data rate is roughly decreased by the duty-cycle factor and a packet should wait until the next forwarder wakes up to transmit it. How to provide high throughput and short delay, while still keeping low power consumption (so low duty-cycle), is the main challenge of the current WSN MAC protocol research. In fact, from typical application point of view, in addition to low rate periodic traffic, burst traffic is triggered following event detections. So an ideal MAC protocol should be able to self-adapt its offered bandwidth to cope with the dynamic traffic load, so that the energy is only used for carrying the application traffic whenever needed.

Lots of low power traffic adaptive MAC protocols have been proposed [5]. For instance, in contention-based WSN, for better dealing with the collisions during burst traffic period, Strawman MAC protocol [9], upon detection of collisions at the receiver, uses a kind of black burst mechanism [14] for better resolving collisions, instead of using random backoff of CSMA. Although Strawman MAC improves the throughput to a certain extent (and does better than RI-MAC [15]), it still introduces overheads and additional delays. So there is space for improvement. To radically resolve this problem, the most efficient way is to only keep CSMA in light traffic for its flexibility, and use TDMA during heavy or burst traffic load periods for solving the inherent drawback of the contention-based MAC, achieving thus high throughput. The first tentative of hybrid CSMA/TDMA for duty-cycled WSN includes IEEE 802.15.4 beacon-enabled mode (CSMA during CAP and GTS during CFP) and Z-MAC [12]. But the basic design approach of Z-MAC is essentially TDMA-based. CSMA is only used when the slot owner has no data to transmit (slot stealing). So only the slot owners can have their pre-allocated bandwidth guarantee. Burst traffic of the other nodes still cannot be efficiently carried. To deal with this problem, adaptive time slot assignment is the best way. TRAMA [11] and AI-MAC [3] follow this idea. But TRAMA suffers from time slot spatial reuse problem and has high overhead for executing the adaptive election algorithm. AI-MAC only relies on the sink initiated query in a tree topology.

In this paper, we present iQueue-MAC, which runs in CSMA in light load. When load increases, the senders' queue length will be used to dynamically allocate time slots to the senders (TDMA). The whole network is composed of two kinds of nodes: simple nodes (e.g. RFD of IEEE 802.15.4) and routers (FFD of IEEE 802.15.4, such as coordinators). Simple nodes only wakeup when they have data to send, so their energy consumption is minimized. Each simple node is associated to a router. A router is responsible of collecting the data packets of the simple nodes that are associated to it. The design of iQueue-MAC protocol follows five key features: using queue-length piggybacking as accurate load information without additional overhead, dynamically allocating TDMA time slots to simple nodes (data senders) for allowing high throughput, using LPL (as X-MAC [10]) to "synchronize" neighboring routers, sending as a burst the queued packets (as T-MAC [16]) from router to router for shortening the channel access delay, and using multi-channels (local channel to a router, and common channel between routers) for exploiting both time and spatial reuse. This results in a highly efficient MAC protocol that we have implemented on STM32W108 chips that offer IEEE 802.15.4 standard communication. For comparison, we also implemented the often referenced RI-

MAC on the same platform, as well as CoSenS [8], which was our previously proposed traffic adaptive protocol. Extensive measurements show its outstanding throughput performance for dealing with burst traffic, while it can still maintain low duty-cycle under light traffic load. Moreover, its traffic adaptation is faster than the existing protocols.

II. RELATED WORK

A lot of protocols have been proposed for variable traffic in WSNs. RI-MAC [15] is a receiver-initiated duty-cycled MAC that aims at handling variable traffic. Receiver nodes frequently wake up and send out probing beacons for polling possible incoming data without using LPL preamble packets. Thus, compared to LPL, RI-MAC enhances throughput by reducing medium occupation. However, RI-MAC suffers from packet collisions when there are multiple senders. Thus throughput is bounded during bursts or heavy traffic. To this end, Strawman MAC [9] schedules transmissions when collisions are detected using extra Collision packets. On the other hand, iQueue-MAC uses a hybrid CSMA/TDMA mechanism to allocate more slots when needed and transform most of the transmission from the random backoff CSMA period into the slotted TDMA period. Z-MAC [12] also adopts a hybrid CSMA/TDMA mechanism, but using a fixed TDMA schedule.

RC-MAC [4] uses a scheduling method to reduce collisions like iQueue-MAC. RC-MAC schedules packet transmissions by ordering the next packet sender using ACK piggybacking. However, how to allocate bandwidth among nodes is not specified in RC-MAC. iQueue-MAC automatically distinguishes the bandwidth demands by reading the sender queue length piggybacked on the data, and enhance the throughput rapidly by allocating extra transmission slots. To support burst packet transmission, BurstMAC [13] also allocates slots for burst transmissions. Senders in BurstMAC apply for slots using specific polling packets. However, BurstMAC requires global synchronization, and the maximum number of two-hop neighbors is also bonded, no more than the number of available radio channels. iQueue-MAC requires only local beacon synchronization and does not limit the number of neighbor senders. Moreover, by piggybacking, iQueue-MAC realizes resource inquiring and slots allocation implicitly in a simple and effective way.

Another popular solution to adjust bandwidth for variable traffic is to prolong the active period in a frame cycle. Max-MAC [6] switches the duty-cycle based on threshold values. When a node senses that the packet receiving rate has gone above a defined threshold value, it doubles its duty-cycle for absorbing more possible traffic. Compared to MaxMAC, iQueue-MAC reacts to traffic variation much more rapidly. A router in iQueue-MAC allocates slots to intensive senders at the start of packet queues build up.

pTunes [19] is a framework that adapts the MAC protocol parameters at runtime for maintaining network performance metrics. pTunes treats MAC parameters tuning as a multi-objective optimization problem and builds up a model of the network. However finding an accurate model itself for the network is not easy, while pTunes still requires extra operations like information collection and dissemination for the optimization which may induce extra overhead.

In our early work, CoSenS [8], a collecting then sending burst protocol was proposed to deal with traffic adaptation. We distinguish simple nodes from routers. A router operates similarly as IEEE 802.15.4 beacon-enabled mode. It dynamically adjusts its data-collecting period (active CSMA period) according to the estimated traffic load. The estimation algorithm is based on the weighted exponential average (similar to that used for RTT in TCP protocol). Moreover, burst transmission mode (similar to T-MAC) is used to transmit packets from router to router. Compared to iQueue-MAC, its traffic estimation algorithm only gives average load, while iQueue-MAC gives exact queue length.

Queue-MAC [18], our former work was proposed to tackle the limited throughput issue in duty-cycled MAC protocols. We have proved that Queue-MAC effectively provides sufficient throughput in a wide range of traffic loads, especially in critical burst periods. The first implementation of Queue-MAC showed a very interesting potential but some aspects have not been taken into account. Namely, multi hop forwarding has not been integrated.

In this paper, we propose iQueue-MAC, the improved Queue-MAC, using the main mechanism of Queue-MAC while integrating the scheme of ”first collect then send burst” of CoSenS. Designed for multi-hop WSNs, iQueue-MAC is an energy-efficient solution for providing immediate and sufficient bandwidth in varied traffic loads, as we will further detail in following sections.

III. IQUEUE-MAC DESIGN

iQueue-MAC mainly targets data collecting networks that often adopt tree routing structure (but not limited to). Devices in the network are classified as routers and simple nodes. We first present the basic idea of iQueue-MAC and then elaborate on different aspects of the design.

A. Basic Idea

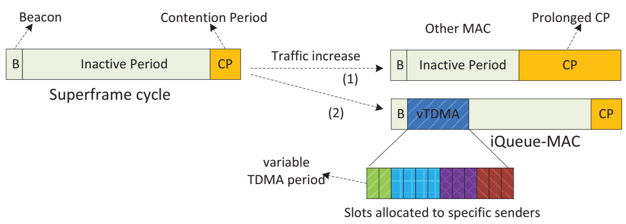

Fig. 1: The basic idea of the iQueue-MAC. Method (1) demonstrates a common way of prolonging the contention period used by most existing duty-cycle MAC protocol. In our iQueue-MAC, as shown in method (2), a variable TDMA period and a CSMA period are integrated to handle adaptive traffic.

Figure 1 shows the basic idea of iQueue-MAC. Most existing duty-cycle MACs schedule an access window, the contention period (CP) in Figure 1, in the superframe cycle for simple nodes to transmit their packets [16][17]. When traffic grows, existing solutions prolong the CP period to transmit more data (method (1) in Figure 1). However, as traffic loads may vary, it is difficult for the transmitter to optimally determine the length of the CP period, and also difficult for the receiver to determine how long it should stay awake. Moreover, as traffic loads increase, contention becomes more and more serious as multiple senders may compete for accessing the channel at the same time. Similarly, a pure TDMA with fixed time slots is also not suitable for traffic adaptive network as it wastes energy for idle-listing in low traffic situation, and suffers large delay for data transmission in heavy traffic situation.

By leveraging the advantages of CSMA and TDMA, iQueue-MAC integrates a variable TDMA period and a CSMA period for energy efficient and traffic adaptive data transmission. The basic idea of iQueue-MAC is shown in Figure 1. Instead of simply prolonging the CP, iQueue-MAC allocates an extra variable TDMA period (vTDMA) within the inactive period to enhance throughput. Simple nodes first apply in CP for extra transmission slots if they have pending packets. Then a router allocates requested number of slots to those senders in the vTDMA period. To realize this in an energy-efficient way, a sender node piggybacks its queue length value onto every data packet. The router checks the queue length information upon receiving a data packet. If the queue length value is non-zero, the router allocates the corresponding slots to the specific sender in the next cycle. In this way, iQueue-MAC dispatches packets transmissions in the TDMA phase as soon as packet queuing starts to build up, leading to short queue length and short packet delay. Moreover, iQueue-MAC has high energy-efficiency by transforming most of the communication into a slot-organized TDMA round.

In the following, we will elaborate on the key issues of designing iQueue-MAC: packet modification, the superframe structure and slot allocation.

B. *Packet Modification*

To enable slots allocation, we slightly modify the packet structure at the MAC layer. As shown in Figure 2, a one byte field called queue indicator is set as the first byte of the packet payload without violating the existing IEEE 802.15.4 standard [7].

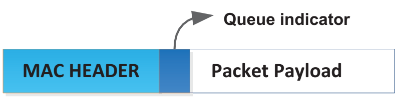

Fig. 2: Packet structure for iQueue-MAC

The value of the queue indicator equals the number of buffered packets currently in the sender's forwarding queue. By examining the value of the queue indicator, any receiver of the data packet can know how many packets the sender still holds for further transmission and thus enables accurate bandwidth allocation. For instance, node_1 has 5 packets in its queue and sends one packet to router A successfully, carrying a queue indicator of 4. After extracting the queue indicator, router A knows that node_1 has 4 packets to be sent and allocates 4 slots accordingly to node_1 to send its pending packets in the next frame cycle.

C. *Superframe Structure*

Routers in the network broadcast beacons independently to divide time into repeating superframes. As depicted in Figure 3, a superframe comprise beacon period, Subframe period, CP and Transmitting Period (TP).

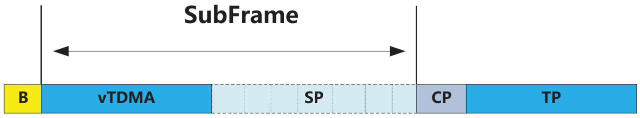

Fig. 3: Superframe structure for iQueue-MAC

Subframe consists of two parts: vTDMA (variable TDMA period) and SP (Sleeping Period). The vTDMA period consists of slots allocated to node devices while SP occupies the rest of Subframe period. During SP, a router sleeps most of the time but still wakes up periodically according to a predefined wake-up interval to check for potential preamble packets from other routers (dotted lines in the SP period). When there is no traffic, Subframe consists of only SP. When traffic grows, more slots will be allocated accordingly, resulting in longer vTDMA and shorter SP. To avoid beacons from colliding with nearby routers, before a new superframe cycle begins, a router sets the Subframe duration to a random value with a predefined average S and varying between 0.5*S and 1.5*S, similar to RI-MAC.

CP follows the Subframe and works as the access window for nodes. Nodes that have pending packets contend for transmission using CSMA/CA mechanism in CP. With the piggybacked queue indicator, the role of the sent out packets in CP is twofold: on the one hand, it is a normal data packet; on the other hand, it informs the router to allocate slots for pending packets. To reduce the contention in CP, every node is restricted to send at most one packet in this period, with remaining packets kept in the queue for the following vTDMA phase. Packets that arrive in the meanwhile are held for the next superframe.

The duration of the CP period is also variable as in T-MAC [16]. Each router adjusts its CP to cope with the end of channel activity by extending CP duration to a certain amount upon every packet reception. Theoretically, CP has a minimum length L as:

$$L = 2 * L_{cca} + L_{CW} + L_{data} \tag{1}$$

where L_{cca} is the time for a CCA execution, L_{CW} represents the largest contention window, and L_{data} represents the time for an intact data packet reception (Data+ACK).

A router burst transmits collected packets to the next hop in TP using LPL (low power listening) technique [2][10]. When the CP ends, the router first sends out a sequence of preamble packets as in X-MAC [2] to inform the next hop router. The first preamble packet is sent out using CSMA for collision avoidance while other following preambles are not. Once acknowledged by preamble-ACK, the router transmits all its buffered packets to the receiver in a burst. The beacon of the next cycle strictly follows the data stream as the router still occupies the medium. In case a router collects no packets in a superframe, there will be no TP period as the router has no packet to forward. Then the router broadcast its next beacon using CSMA/CA for collision avoidance.

D. Slots Allocation

Considering an extremely light traffic case where very few nodes report data, which is also a common scenario in WSNs, the CP period can handle all the scattered packets as the traffic load is low, Also, as no slots are allocated, there is no vTDMA period in the Subframe. Both routers and nodes sleep most of the time to save power. In this way, iQueue-MAC achieves a very low idle traffic duty-cycle.

When a sudden event is detected, nodes in the surrounding area may turn into intensive senders simultaneously and cause the traffic to grow significantly. Packet queuing starts to occur due to limited throughput. And by extracting the piggybacked queue indicator byte from every received data packet, a router knows exactly how many slots should be allocated to each sender to serve the increasing traffic load.

Every router maintains two lists: an ID list recording IDs of senders that have been allocated slots, and a Slot Allocating list for recording the numbers of the allocated slots. A router assesses slots allocation upon every packet reception, removes the node's ID from ID list if the queue indicator instructs 0, or adds more slots if needed. Together with other information, the router merges the two lists into the beacon before sending it out, as shown in Figure 4.

Header	ID list	Slot Allocation list

Fig. 4: Beacon structure for iQueue-MAC

Upon receiving a beacon, a node gets synchronized with the router's new superframe and checks for allocated slots and their locations within the Subframe. If the node's ID is in the ID list, the node finds out its sequence K in the ID list for further determining the number of allocated slots $N_{slots}[K]$ and the starting time of its slots period $T_{start}[K]$:

$$N_{slots}[K] = Slot\_Alocation\_list[K] \tag{2}$$

$$T_{start}[K] = Size_{slot} * \sum_{i=1}^{K-1} N_{slots}[i] \tag{3}$$

where $Size_{slot}$ is the size of a TDMA slot. Then the node sleeps until $T_{start}[K]$ and transmits its pending packets back to back in its allocated slots. In case that the node finds no slots are allocated to it, it continues trying to send its packet during CP. On the other hand, if the node still finds itself with pending packets at the end of its slots period, it skips the contention in CP by knowing that it should have been allocated slots in the next superframe cycle. This execution further relieves the contention in CP by ticking out an additional number of contenders.

Figure 5 shows an example run of iQueue-MAC. Both N1 and N2 have multiple data packets to send. After receiving the router's first beacon, each of them sends out one packet during the CP of the first superframe after the generation of the packets and then retreat from CP. By checking the queue indicator of the received data packet, router R schedules 3

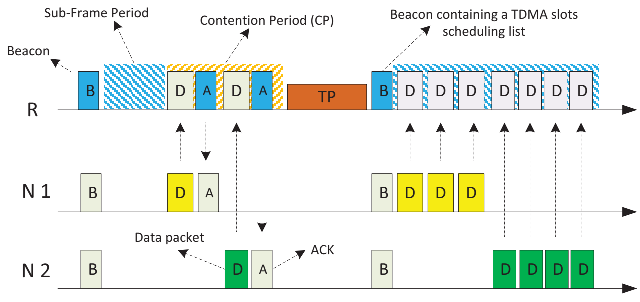

Fig. 5: An example run of iQueue-MAC

slots for N1 and 4 slots for N2 in the next cycle. Each node waits for the second beacon that contains a schedule list, finds the allocated slots period and sends out all the scheduled packets in a burst.

Slot Allocation Strategy: Assuming that one TDMA slot is just sufficient for one single packet transmission with ACK confirm, we use a flat strategy in slots allocation: the number of allocated slots for a node is proportional to its current queue indicator value. For a given Subframe duration , the number of available slots N is bounded to a maximum value M:

$$N = \sum_{i=1}^{k} Slot\_Alocation\_list[i] <= M = \frac{L_{sub-frame}}{Size_{slot}} \tag{4}$$

The router stops scheduling new slots once N is equal to M and k represents the number of nodes with queued packets.

Multi-channel Operation: iQueue-MAC utilizes multi-channel technique in the vTDMA period to allow parallel transmissions of the scheduled packets among neighbor routers. We assume that a channel scheduling algorithm allocates non-overlapping sub-channels to different routers in a local area. The beacon contains the router's sub-channel sequence and each node switches to the defined sub-channel before transmission in the vTDMA period. After the vTDMA period, all nodes and the router switch back to the public channel, and all the other communications will be carried out on the public channel. However, as the first step of our work, in this paper, we did not try to tackle the channel allocation problem. In fact, this has been set as one goal of our future work of implementing an integrated version of iQueue-MAC. Actually, in the following implementation experiments, we predefine non-overlapping sub-channels to different routers.

E. Discussion

By using the hybrid CSMA/TDMA mechanism, we can deal with varied traffic conditions in an efficient way. When the network is with light traffic, it is sufficient to use CP to handle all the rarely generated packets. As traffic grows, iQueue-MAC enhances its throughput immediately by allocating extra slots into the Subframe period. CP actually absorbs only a small part of the traffic load and mostly acts as a signaling mechanism to inform the router to setup the following TDMA period, which will handle most of the transmissions. As almost all the slots are allocated based on queue-length requests, no slot is wasted.

Thus iQueue-MAC maintains high energy-efficiency. On the other hand, iQueue-MAC allocates slots as soon as buffering emerging, thus achieving immediate throughput enhancement and guaranteeing short packet delay.

IV. IMPLEMENTATION

We have successfully implemented iQueue-MAC on IEEE 802.15.4 standard STW32W108 chips [1], and conducted numerous experiments for intensive evaluation. For comparison, we also implemented CoSenS and RI-MAC, and extended RI-MAC with multi-channel operation. We refer to this RI-MAC implementation as RI-MAC-MC. Like iQueue-MAC, every router in RI-MAC-MC is assumed to have a non-overlapping sub-channel sequence. Except for the first beacon broadcast on the public channel, a RI-MAC-MC router guides the following communication procedure onto its sub-channel. All devices turn back to the public channel at the end of a communication cycle. As RI-MAC-MC follows the main scheme of RI-MAC, it is still a contention based protocol and does not support back to back transmissions as multiple senders may simultaneously compete for medium accessing. CoSenS does not use multi-channel technique.

For iQueue-MAC and RI-MAC-MC, we assume the existence of a channel allocation algorithm at the initiating stage of the network. However, at this point we actually predefine non-overlapping sub-channels to different routers.

To explore iQueue-MAC's performance under different scenarios, we set up two sample experiments and a general one, all of them on real world test-beds. In the first sample experiment, we test the performance of the different MACs in a wide range of traffic loads, while in the second sample experiment, we measure how different MACs react to sudden traffic bursts. We simulate a real-world general application scenario with a 46 nodes test-bed in the last experiment. Key MAC parameters are shown in Table I.

TABLE I: Parameters Setting For MAC Protocols Evaluation

	iQueue-MAC	CoSens	RI-MAC-MC
Mean Subframe	500ms	500ms	500ms (sleep interval)
Minimum CP	15ms	6ms	-
Slot size	5ms	-	-
Max retry	5 (in CP)	5 (in WP)	5
Multichannel	Yes	No	Yes

Data packet size is always 120 bytes, and the slot size in the experiment is set to 5ms to grant one packet transmission with ACK. An actually larger minimum CP value is used in iQueue-MAC as we found that it seems to yield the best performance in numerous experiments. Available IEEE 802.15.4 channels from channel 11th to channel 21th are used in iQueue-MAC and RI-MAC-MC.

V. EXPERIMENTAL RESULTS

A. Adapting To Varying Traffic Loads

We set up a simple test bed that contains one router and 10 simple nodes as depicted in Figure 6. Simple nodes are closely placed and each node continuously generates up to 500 data packets with Poisson random distribution and sends them to the router. The router further relays those packets to the sink node. Each experimental run lasts for 800 seconds. By varying the node data rate from 1packet/1500ms (0.08kBps) to 1packets/100ms (1.2kBps), we measured the performance of the three MACs.

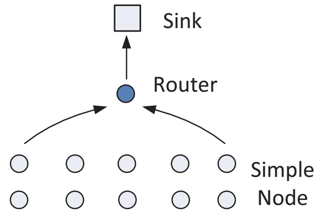

Fig. 6: The topology of our testbed for performance evaluation

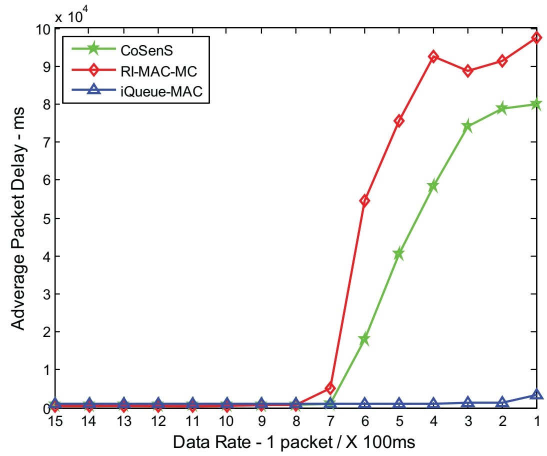

Fig. 7: The effect of data rate on the average delay

Figure 7 shows the average packet delay over different traffic conditions. At the beginning, all MACs have short delay under light traffic. When the data rate exceeds 1packet/800ms, RI-MAC-MC and CoSenS start to suffer from notably increasing delay. iQueue-MAC successfully maintains low packet delay over all scenarios.

Figure 8 shows the packet reception ratio comparison of the first sample experiment. Each experimental run consists of generating 5000 packets. All MACs maintain high packet reception ratio under light traffic. When the data rate exceeds 1packet/600ms, a large number of packets are dropped in CoSenS and RI-MAC-MC due to limited throughput, as most nodes have already reached their maximum queue-length (200). When the network is with high traffic, a single inquiring beacon in RI-MAC-MC causes fierce contention as it wakes multi nodes to compete the channel simultaneously in every round, which is an inherent drawback of the protocol. We have also observed on the sniffer that packets from different nodes continuously collided with each other and the contention window got increasingly extended but still failed to mitigate the contention and retransmissions. Similar to RI-MAC-MC, CoSenS suffers from heavy retransmissions and packet loss as most of the nodes compete for transmission during the same contending period. On the contrary, iQueue-MAC succeeds in absorbing almost all the generated packets, proving that

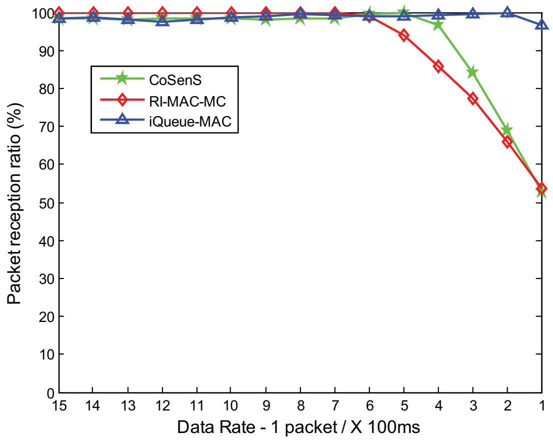

Fig. 8: The effect of data rate on the packet reception ratio

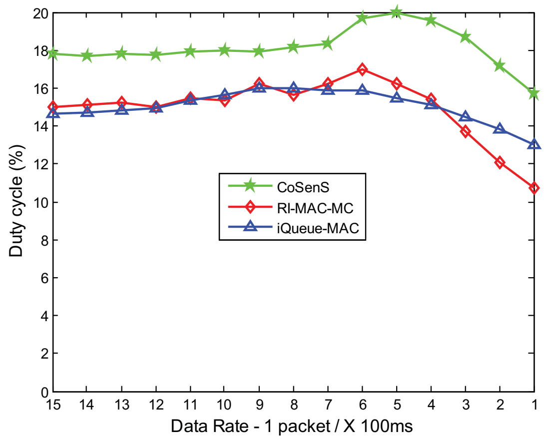

Fig. 10: The effect of data rate on the duty-cycle

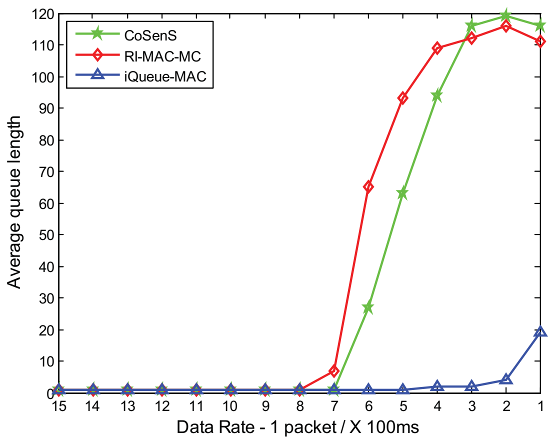

Fig. 9: The effect of data rate on the average queue length

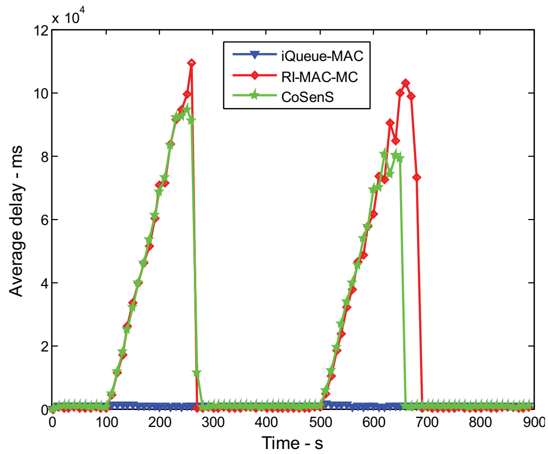

Fig. 11: The comparison of the average delay

iQueue-MAC is robust to varied traffic.

Figure 9 shows the average queue length of nodes in different MACs. iQueue-MAC succeeds in maintaining very short queue-length over all experimental scenes. That is due to the fact that iQueue-MAC suppresses queuing by allocating the corresponding slots. CoSenS and RI-MAC-MC, however, failed to provide enough bandwidth as traffic grows, resulting in substantial packet accumulation and long queue lengths, which also explain the larger delays in Figure 7 and the higher packets loss in Figure 8.

Figure 10 is about the duty-cycle that every MAC achieves. As we can see, CoSenS is less energy-efficient than iQueue-MAC and RI-MAC-MC, both of which have similar performances. For all MACs the duty-cycle drops as the data rate grows beyond 1packet/600ms. Since under high traffic load RI-MAC-MC and CoSenS drop many packets due to queue overflows, the result is a lower number of packets transmitted and thus a shorter active time. On the other hand, as higher traffic loads usually lead to longer queue lengths, iQueue-MAC arranges more slots in the vTDMA period when data rate grows. As the scheduled TDMA transmission is more efficient than contention based transmissions, iQueue-MAC thus consumes less active time during high traffic period, and achieves higher efficiency.

B. *Reacting To Burst Traffic*

We used the same test-bed as described in section 5.1 and conducted a new set of experiments to show how different MACs react to peak or burst traffic. Initially, all nodes generate packets under a low data rate of 1packet/5seconds (0.024kBps). Two bursts are set to occur at 100s and 500s respectively and both last for 50s. During the burst period, all nodes adopt a higher data rate of 5packets/second (0.6kBps). We measured key metrics along with the time.

Figure 11 shows the network average packet delay. At the beginning, all MACs have low latency. Then, when burst occurs, RI-MAC-MC and CoSenS start to suffer from sharply increasing delay. The situation continues to deteriorate until all buffered packets are dispatched. Compared to the other MACs, iQueue-MAC maintains its low latency even under the burst period. Figure 12 reveals the average queuing situation in all MACs which also explains the results in Figure 11. It clearly shows that iQueue-MAC outperforms the other two MACs in maintaining a short queue length. This is because iQueue-MAC

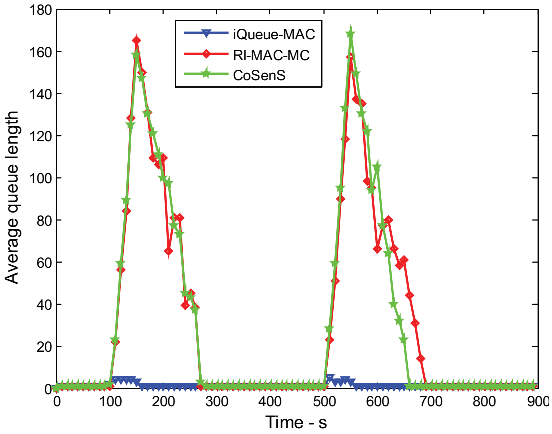

Fig. 12: The comparison of the average queue length

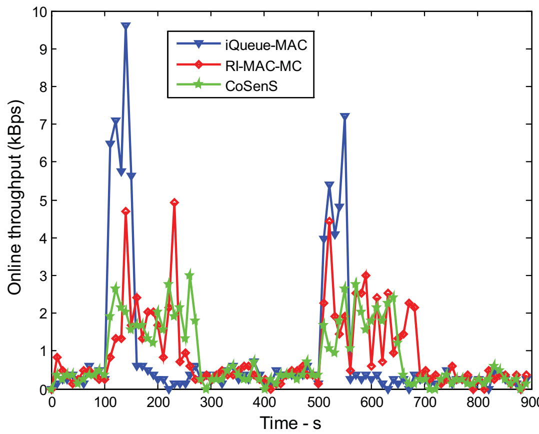

Fig. 13: The comparison of the online throughput

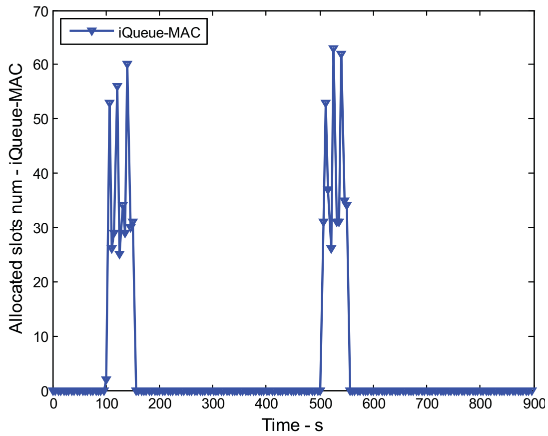

Fig. 14: The number of allocated Slots in iQueue-MAC

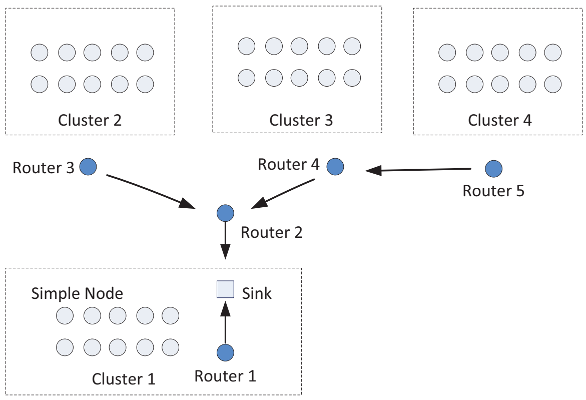

Fig. 15: The topology of the general experiment

can stop packets from accumulating by providing sufficient extra bandwidth. RI-MAC-MC and CoSenS, on the other hand, are vulnerable to traffic bursts. Their bounded throughput leads to large packets queuing.

Figure 13 shows the measured throughput of each MAC. We clearly see that iQueue-MAC sharply enhances its throughput at the emergence of a burst period, while RI-MAC-MC and CoSenS can only provide about one fourth of the throughput of iQueue-MAC and spread over a wider period. As a result, iQueue-MAC eliminates queuing and performs much better than the other MACs. The key of the bandwidth enhancement of iQueue-MAC is highlighted in Figure 14, which is the sudden allocation of TDMA slots. As we can see, iQueue-MAC allocates large numbers of slots during the burst periods. These allocated slots contribute to higher instantaneous bandwidth and thus shorter packet delay. Compared to other solutions, iQueue-MAC uses queue length as the feedback value for taking further control actions. This allows iQueue-MAC to react much faster as it does not need to predict the traffic loads.

C. General Experiment

As depicted in Figure 15, we set up a more general testing environment for simulating a real-world multi-hop application scenario. A test-bed of 46 nodes scattered over one layer of the lab building and arranged into a 4 clusters network contains at most 4-hops transmission distance. Each cluster is placed in a lab room and contains 10 children-nodes and one router (parent) node. Data relaying paths are also fixed to factor out routing influence on experimental results. Initially, all children-nodes generate packets under a light data rate of 1packet/5seconds (0.024kBps). Then, to simulate a series of urgent events, each cluster will experience a burst period adopting a higher data rate of 2 packets/1s (0.24kBps), as shown in Table II. To evaluate a more critical scenario in the network, all children-nodes simultaneously switch to the burst state during the final burst. The whole experiment lasts for 900 seconds.

TABLE II: Burst Periods During General Experiment

Cluster ID	Cluster_1	Cluster_2	Cluster_3	Cluster_4	All Clusters
Burst period	100s-120s	200s-220s	300s-320s	400s-420s	600s-620s

Figure 16 shows the average packet delay of the general

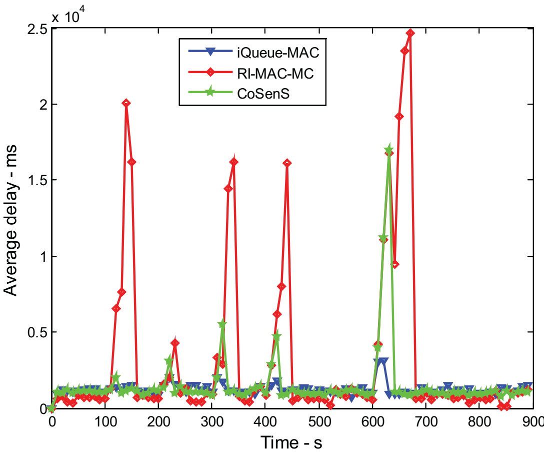

Fig. 16: The comparison of the average delay in the general experiment

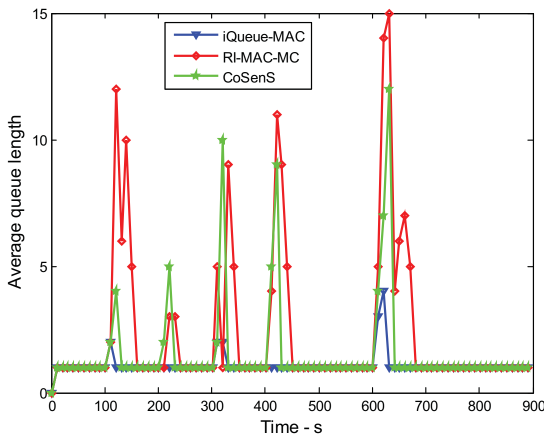

Fig. 17: The comparison of the average queue length in the general experiment

experiment. RI-MAC-MC has the worst performance during each burst period and CoSenS performs better, while iQueue-MAC behaves the best. iQueue-MAC maintains low packet delay over the whole experiment. Figure 17 shows the buffering situations with all MACs. Again, iQueue-MAC succeeds in maintaining much shorter queue lengths than other MACs. This is achieved since iQueue-MAC allocates a sufficient number of slots to senders as soon as packets queuing is detected. Conversely, CoSenS and RI-MAC-MC lead to relatively longer queues during the burst periods as they react slower to the traffic changes.

Figure 18 shows that iQueue-MAC provides faster and stronger throughput enhancement when burst traffic occurs. Higher throughput leads to faster packets dispatching thus iQueue-MAC also achieves shorter delay and queue length as depicted in Figure 16 and Figure 17, respectively. On the other hand, the slower reactions in throughput adaptation in CoSenS and RI-MAC-MC result in worse performance.

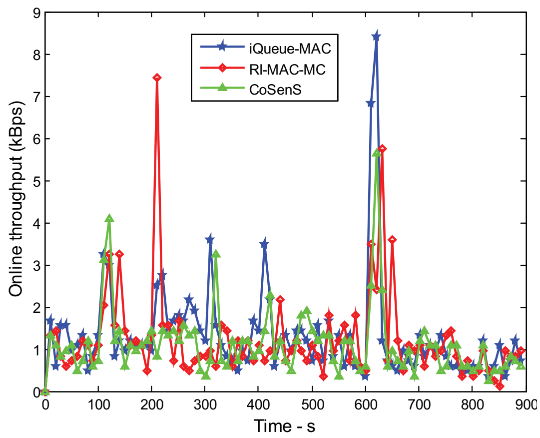

Fig. 18: The comparison of the online throughput in the general experiment

VI. CONCLUSIONS

In this paper, we presented iQueue-MAC, a MAC protocol for WSNs that efficiently manages the RF wireless medium resource under varying traffic conditions. This management combines CSMA and TDMA types of medium access control, the former for light traffic conditions and the latter for heavier traffic. The crux of the protocol is an accurate and fast assessment of the instantaneous communication requirements that promptly detects packets queuing in the senders and allows the router to schedule sufficient TDMA slots. Such assessment is carried out piggybacking a queue indicator onto all data packets, with slots requests and allocation done implicitly. Packets queuing is mitigated by allocating slots right upon queuing detection.

Compared to other solutions, iQueue-MAC has the following advantages: first, iQueue-MAC provides a bandwidth that precisely matches the requirements as it can learn the traffic load more accurately and allocate the needed slots; second, iQueue-MAC maintains high efficiency as it mitigates contention and retransmission by using TDMA slots for intensive senders.

We have successfully implemented iQueue-MAC on STM-32W108 SOC chips, and for comparison we also implemented CoSenS and RI-MAC-MAC and tested their throughput, network delay and queues length in numerous experiments. The results clearly show that iQueue-MAC outperforms CoSenS and RI-MAC-MAC in all the experimental scenarios.

ACKNOWLEDGMENTS

This work was supported in part by the French National Research Agency (ANR) and the National Science Foundation of China (NSFC) Joint Quasimodo Project (ANR 2010 INTB 0206 01, NSFC 61061130563), in part by the National Natural Science Foundation of China (NSFC) under Grant (61273079), in part by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant (XDA06020300, XDA06020201), in part by State Key Laboratory of Industrial Control Technology of Zhejiang University under Grant (1206, 1207), and in part by Key Laboratory of Wireless Sensor Network & Communication of Chinese Academy of Sciences under Grant 2011001.

REFERENCES

[1] Stm32w108 datasheet. `www.datasheetarchive.com/STM32W108-datasheet.html`.

[2] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proc. of the 4th International Conference on Embedded Networked Sensor Systems*, pages 307–320, 2006.

[3] S. Chatterjea, L. Van Hoesel, and P. Havinga. AI-LMAC: An adaptive, information-centric and lightweight MAC protocol for wireless sensor networks. In *Proc. of the 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference*, pages 381–388, 2004.

[4] P. Huang, C. Wang, L. Xiao, and H. Chen. RC-MAC: A receiver-centric medium access control protocol for wireless sensor networks. In *Proc. of the 18th International Workshop on Quality of Service*, pages 1–9, 2010.

[5] P. Huang, L. Xiao, S. Soltani, M. Mutka, and N.Xi. The evolution of mac protocols in wireless sensor networks: a survey. *Communications Surveys & Tutorials, IEEE*, pages 1–20, 2012.

[6] P. Hurni and T. Braun. Maxmac: A maximally traffic-adaptive mac protocol for wireless sensor networks. *Wireless Sensor Networks*, pages 289–305, 2010.

[7] IEEE 802.15.4 Standard. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). *IEEE: Piscataway*, NJ, USA, 2006.

[8] B. Nefzi and Y.-Q. Song. QoS for wireless sensor networks: Enabling service differentiation at the MAC sub-layer using CoSenS. *Ad Hoc Networks*, 10(4):680–695, 2012.

[9] F. Österlind, L. Mottola, T. Voigt, N. Tsiftes, and A. Dunkels. Strawman: resolving collisions in bursty low-power wireless networks. In *Proc. of the 11th International Conference on Information Processing in Sensor Networks*, pages 161–172, 2012.

[10] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proc. of the 2nd International Conference on Embedded Networked Sensor Systems*, pages 95–107, 2004.

[11] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *Wireless Networks*, 12(1):63–78, 2006.

[12] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu. Z-MAC: a hybrid MAC for wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 16(3):511–524, 2008.

[13] M. Ringwald and K. Romer. BurstMAC–An efficient MAC protocol for correlated traffic bursts. In *Proc. of the 6th International Conference on Networked Sensing Systems*, pages 1–9, 2009.

[14] J. L. Sobrinho and A. S. Krishnakumar. Quality-of-service in ad hoc carrier sense multiple access wireless networks. *Selected Areas in Communications, IEEE Journal on*, 17(8):1353–1368, 1999.

[15] Y. Sun, O. Gurewitz, and D. B. Johnson. RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In *Proc. of the 6th ACM conference on Embedded Networked Sensor Systems*, pages 1–14, 2008.

[16] T. Van Dam and K. Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proc. of the 1st International Conference on Embedded Networked Sensor Systems*, pages 171–180, 2003.

[17] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proc. of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1567–1576, 2002.

[18] S. G. Zhuo, Y.-Q. Song, Z. Wang, and Z. B. Wang. Queue-MAC: A queue-length aware hybrid CSMA/TDMA MAC protocol for providing dynamic adaptation to traffic and duty-cycle variation in wireless sensor networks. In *Proc. of the 9th IEEE International Workshop on Factory Communication Systems (WFCS)*, pages 105–114, 2012.

[19] M. Zimmerling, F. Ferrari, L. Mottola, T. Voigt, and L. Thiele. pTunes: Runtime parameter adaptation for low-power MAC protocols. In *Proc. of the 11th international conference on Information Processing in Sensor Networks*, pages 173–184, 2012.