A High Throughput MAC Protocol for Wireless Sensor Networks in Surveillance Applications

Jian-hua WANG and Yan YU *
Institute of Computer Science and Information Engineering, Harbin Normal University, Heilongjiang, Harbin, 150025, China
Email: WJH@vip.163.com, Yuyan9999@vip.qq.com

Madukasi Charles Nnaemeka
God's time Contracting Company, Federal Housing Estate, Onitsha, Anambra State, 430273, Nigeria
Email: cmakas4all@yahoo.com

Abstract—Monitoring a given environment is a kind of major applications in wireless sensor networks. These WSNs should often meet special requirements, such as high throughput support, service differentiation support and energy efficiency. However, related works always emphasis on one or two of them, and hardly consider comprehensively. In this paper, we propose a new-style MAC protocol, which based on the above-mentioned factors. Cluster-based multi-hop scheduling and priority-aware schedule switching are crucial technologies in the MAC protocol. Moreover, idle listening energy consumption is reduced by using a synchronized duty cycle. Experiments show that the proposed strategy achieves our goals. The energy consumption of the radio module is reduced while high throughput is provided.

Index Terms—WSN; High Throughput; MAC; Cluster-Based Scheduling; Priority-Aware Switching

I. INTRODUCTION

One of the key applications of wireless sensor networks (WSNs) is monitoring a given environment, such as surveillance of enemy invasion in security systems [25]. In WSNs, thousands of battery-powered sensor nodes generate management packets periodically to maintain the network or data packets unexpectedly when an event is detected. Both types of packets should be delivered to the sink within the given time deadlines, and it is especially critical that the event must not be missed or informed after too long a delay. These WSNs should meet the following three requirements.

High throughput support: In surveillance networks (such as enemy invasion), the sensed data reflect the target status, such as the position, pictures of moving targets or the temperature of a fire. The target status can change easily with time. Thus, the sensed data is only valid for a limited period of time (i.e., packet deadline [19], [25]). WSNs should deliver packets to the sink within their deadline (e.g., five seconds) as many as possible. This indicates that providing high throughput, which is defined as the number of packets that arrived at the sink node within their deadline during a unit time [14], is an important requirement of WSNs.

Service differentiation support: In such surveillance networks, the importance of a data packet depends on its types [9], [13]. For example, the packets for the detected enemy have a higher priority than routine management packets. WSNs should preferentially deliver high-priority packets.

Energy efficiency: Since sensor nodes are usually deployed in places where people cannot easily access, the nodes should operate energy efficiently to increase the battery lifetime [1], [3].

This paper proposes a new MAC protocol which provides energy efficiency and service differentiation while maintaining high throughput. Before presenting our work, we briefly describe several previous works that are closely related to our work.

SMAC [1] and TMAC [2] first achieve energy efficiency by suggesting the synchronous duty cycle. Duty cycle is the periodic listen (i.e. turning on the radio) and sleep (turning o ff the radio) of each node. In the listen period, all sensor nodes activate their radio interfaces and participate in data forwarding. In the sleep period, they turn o ff the radio interfaces for energy conservation. By SMAC-like duty cycling, each node conserves much energy and prolongs the battery lifetime. However, as data packets can only be forwarded during the listen period, emergent events cannot be notified within a short deadline and the throughput is sacrificed for energy efficiency.

LASMAC [3], [4], RMAC [5], and DW-MAC [12] are energy efficient as well as achieve relatively higher goodput by node scheduling in a multi-hop environment. Unlike SMAC, the listen period is used to schedule nodes in a routing path (i.e., deciding when to wake up, send, or receive) and data packets are forwarded in the sleep
period along with the scheduled nodes. Other unscheduled nodes sleep for energy saving. As packets are forwarded in the sleep period, such schedule-based protocols support higher throughput than SMAC. However, because only a single path is scheduled for a data flow, the throughput of the previous schedule-based protocols still suffers from retransmissions occasionally happened on each hop in lossy WSNs [6, 19].

Furthermore, such schedule-based protocols have difficulties in providing service differentiation. In LASMAC and RMAC, once a node is reserved for delivering a low-priority packet, an urgent packet (e.g. a highest-priority packet) that arrives late at the node must wait until the low. Priority packet goes through according to the schedule. For this problem, previous service differentiation methods [7]-[9] (e.g. varying DIFS and CW size depending on packet priority) can be applied to LASMAC, but increased service differentiation effect is small. This necessitates a totally new design of service differentiation scheme.

To achieve the goals mentioned above, we propose a MAC protocol which supports high Throughput, Energy efficiency, and Service differentiation (GES-MAC). GES-MAC consists of two techniques: Cluster-based Multhop Scheduling and Priority-aware Schedule Switching.

Cluster-based Multi-hop Scheduling (CMS) is designed to maintain energy efficiency by using a duty cycle and to provide high throughput by solving the retransmission problem on each hop. Unlike previous scheduling-based approaches that schedule only nodes in the routing path, our CMS schedules nodes by the cluster, called scheduling cluster (SC), which consists of both routing nodes and their one-hop neighbor nodes. CMS enables data packets to be forwarded hop-by-hop in the opportunistic forwarding manner [17], [18] during the sleep period. Then, the number of retransmissions can be dramatically reduced because the probability that at least one node in a cluster will successfully receive the packet is much higher than the probability that only the next routing node successfully will receive the packet, resulting in throughput increase. Our experimental results show that our approach can provide high throughput, although a little more energy is consumed than LASMAC.

Priority-aware Schedule Switching (Pass) is proposed to support service differentiation in schedule-based WSNs. Pass allows high-priority packets to preempt the schedule of low-priority packets so that the high-priority packets can be served earlier. Then, even though a node is reserved for a low-priority packet earlier, a high-priority packet is forwarded faster than the low-priority packet, resulting in the higher throughput of high-priority packets. We minimize the overhead in Pass to avoid throughput degradation. Our experimental results show that Pass supports service differentiation in schedule-based WSNs with negligible overhead.

Note that each node in GES-MAC manages only its one-hop neighbors’ information for scalability. Note that this paper just focuses on the usage of duty cycle to enhance energy e and the balancing of energy utilization among sensor nodes is beyond the scope of this paper. It is because the current version of GES-MAC is implemented on geographic routing [28], but it may be achieved by applying energy aware routing protocols [26] to GES-MAC.

II. RELATED WORKS

To reduce energy consumption, SMAC [1] and TMAC [2] suggest the duty cycle, which means periodic listen (i.e., turn on radio module) and sleep (i.e., turn off radio module) sensor nodes. Duty cycle is common feature of energy efficient MAC [3], [5], [6], [10], [15], [16]. As the energy consumption on the radio module is dominant in sensor nodes (more than 60% of total) [1], the energy consumption is reduced. However, in SMAC, data packets are forwarded during only listen period, and data packets can be forwarded just one or two hops in a cycle. Thus, SMAC has throughput degradation problem. For the same reason, a naive combination of SMAC and COR has also the problem.

BMAC [10] and XMAC [11] achieve higher energy efficiency by exploiting shorter listen period (e.g. 1-5% of a cycle) than SMAC. They also enhance the throughput by sending a data in the sleep period. In BMAC, a node sends a long preamble packet for its next node not to sleep in the sleep period, and then, the node transmits the data packet. In this manner, BMAC shows better throughput than SMAC, but the usage of preamble packet induces the delay problem because the data packet goes through the overhead by preamble packets on every hop. Thus, those protocols have difficulties in providing high throughput, especially in a multi-hop environment.

To compensate the throughput degradation caused by the duty cycle, LASMAC [3], [4], RMAC [5], and DW-MAC [12] use the listen period to schedule nodes in a routing path. In the example of Fig. 1, node A sends a LAS-RTS packet through routing nodes B, C, D, and E in the listen period. Then, the nodes are scheduled to wake up at the specific time of the sleep period with the information of LAS-RTS. In the sleep period, only the scheduled nodes wake up based on their schedule and forward data packet while the other nodes sleep in the sleep period. In this manner, LASMAC and RMAC achieve higher throughput than SMAC while maintaining energy efficiency.

However, compared to opportunistic forwarding approaches, such as COR [17], the throughput enhancement is still insufficient. Our experimental results show that the throughput of LASMAC is only 40% that of COR. We analyzed LASMAC and found that the throughput degradation is mainly caused by retransmissions in data forwarding (i.e., in sleep periods). To achieve similar performance to COR without losing the energy efficiency of LASMAC or RMAC, retransmissions should be reduced.

To reduce retransmissions, combining the concept of LASMAC (or RMAC) and COR can be considered. How-ever, combining the concept is not straightforward, because LASMAC needs a routing path, on contrary, COR chooses the best next-hop node on every
transmission dynamically. We easily consider a naive combination of LASMAC and COR, which modifying LASMAC to schedule more neighbors (B1 or C1) of each routing node (B or C) in the listen period and to ask them to be awake, as a solution. However, such a naive combination has a critical problem shown in Fig. 2. During the sleep period, a data packet can be forwarded from node A to one of nodes B and B1 opportunistically, (i.e., reliably), but we cannot ensure reliable transmission from node B1 to the next-hop nodes (C, C1) because the link qualities among them are not considered in scheduling. In the worst case, nodes B1 may have no link to the next-hop nodes. Thus, we cannot expect considerable throughput enhancement from the naive combination of LASMAC and COR (just 50% of COR), necessitating the development of a new approach for scheduling and forwarding to create a synergetic effect.

During the sleep period, a data packet can be forwarded from node A to one of nodes B and B1 opportunistically, but we cannot ensure reliable transmission from node B1 to the next-hop nodes (C, C1) because the link qualities among them are not considered in scheduling. In the worst case, nodes B1 may have no link to the next-hop nodes. Thus, we cannot expect considerable throughput enhancement from the naive combination of LASMAC and COR (just 50% of COR), necessitating the development of a new approach for scheduling and forwarding to create a synergetic effect.

Figure 1. Operation of LASMAC [3], [4] and RMAC [5].

Figure 2. Problem of naive combination of LASMAC and COR

Furthermore, LASMAC and RMAC have difficulties in providing service differentiation. Figure 3 shows the problem of LASMAC. Node Y schedules node D for Data1 (low-priority) first. After that, if a LAS-RTS packet for Data2 tries to schedule node C, node C is already scheduled to NAV. Thus, node C is scheduled for Data2 with time delay, even though the priority of Data2 is higher than that of Data1. In worst cases, even though the same number of high and low-priority packets is generated, the throughput for high-priority packets is lower than that for low-priority packets. Applying previous service differentiation methods [7]–[9] (e.g., by varying DIFS or CW size) is not effective to the problem either. This necessitates the development of an answer vice differentiation method for our environment.

In summary, the technical advancement flow of the aforementioned works is shown in Fig. 4. Previous works or their simple combinations have difficulties in providing high throughput in duty-cycled environments and in providing service differentiation, as shown in Table 1. To achieve the requirements (i.e., providing energy efficiency and service differentiation while maintaining high throughput), we propose a new MAC protocol, GES-MAC.

III. PROPOSED SCHEME

A. Cluster-Based Multi-Hop Scheduling (CMS)

CMS uses a synchronized duty cycle and each cycle is divided into three periods: Sync period, Scheduling period, and Forwarding period. In the Sync period, each node wakes up and shares SYNC packet in order to synchronize the duty cycle of each node [3], [5], [12] and in order to fill the data structure of each sensor node. In the scheduling period, CMS schedules nodes which are needed for data forwarding by unit of cluster, called Schedule Cluster (SC). In the forwarding period, only scheduled nodes keep awake and join data forwarding, and the other nodes sleep for energy conservation. Unlike previous schedule-based approaches [3], [5], [12], CMS dynamically schedules nodes by unit of SC, which is defined as a routing node and its one-hop neighbor nodes. To this end, CMS manages the one-hop Neighbor Table (NT) and uses a control packet called Cluster-based Multi-hop RTS (CM-RTS) packet. In this section, first, we explain how to manage NT, and then, we briefly explain how to schedule SCs and forward data through SCs.

In CMS, each node manages the one-hop Neighbor Table (NT) which contains the location information, the link reliability, Dist Gain, and the neighbor nodes of each one-hop neighbor nodes, etc. One-hop neighbor nodes are the nodes which communicate directly with one-hop broad-cast. The information on NT is easily
gathered by using SYNC packet. First, the location information of a neighbor node is easily gathered by adding the location information on the SYNC packet. In Fig. 5, node i broadcasts its SYNC packet with its location information. Then, its neighbors (e.g., node x and node j) easily know the location information of node. The location information is included in the SYNC packet only initially, because nodes have almost no mobility. Second, the link reliability is got by the SYNC packet success rate. On the SYNC information (SYNC info), SYNC sequential number is included. With the SYNC sequential number, node x knows the total number of transmit-ted packets from node i and also keeps counting the number of the successfully received packets. Then, the link reliability of node i is easily measured by the ratio of two values [22], [23]. Third, Dist Gain is got by calculation. Dist-Gain represents that which neighbor node is closer to the sink node. At node x, the Dist Gain for node j is calculate as:

\[
\text{DistGain}(j) = \text{DistToSink}(x) - \text{DistToSink}(j)
\]

where Dist To Sink(m) is the distance from node m to the sink node. As the neighbor node whose Dist Gain x Link reliability is maximum is selected as the next routing node in CMS (i.e. geographic routing [28]), Dist Gain is managed. Fourth, the neighbor nodes of a neighbor are shared by adding an ID of the broadcaster’s neighbor. In Fig. 5, node i adds either the ID of node x or node j at each SYNC broadcasting. Then, node x knows node x and node j are all neighbor nodes of node i. In the example of Fig. 5, node k is not a neighbor of node i because of the block. In this manner, each node can manage the NT without big overhead.

This effectively reduces the number of retransmissions, because of link characteristics. Basically, the transmissions from the sender to multiple receivers are pair wises independent [17], [20]. Thus, in the example of Fig. 6, if the link re-liability between node A and each node in SC-B is 50% each, the probability that at least one node among nodes B, B1, and B2 will successfully receive the data packet increases to \(1 - (1 - 0.5)^3 = 87.5\%\). In some cases, there can be temporal correlation among some links [24], but the links are independent for most time. Therefore, by arbitrating multiple receivers so that only one of them gets to participate in data forwarding to avoid collision, we can expect high throughput.

Some nodes in a SC (e.g., nodeB3) do not overhear the CM-RTS packet by temporal fading and shadowing. Then, the nodes do not wake up at the forwarding period. Thus, when a forwarder selects the candidates of new forwarder, the probability about this case should be considered.

Note that the role of routing nodes in CMS is different from that in previous schedule-based approaches [3]–[5], [12]. In the previous approaches, routing nodes deliver RTS and data packets through a single routing path. However, in CMS, routing nodes handle only RTS (i.e., CM-RTS) packets to schedule nodes in the scheduling period.

B. Priority-aware Schedule Switching (PaSS)

The primary goal of Pass is to provide service differentiation in schedule-based wireless sensor networks (WSNs). Here, service differentiation support means that increasing the throughput of high-priority packets as many as possible by forwarding high-priority packets preferentially. In achieving the purpose, the major obstacle is the cases that some SCs are already scheduled for a low-priority packet. Previous approaches are not effective in solving the problem cases or result in total throughput degradation, as mentioned in Sect. 2. In this section, we propose a new method that solves the obstacle without much total throughput degradation.

To achieve service differentiation, PaSS allows high priority packets to preempt the schedules of low-priority packets so that the high-priority packets can be served earlier. Then, the throughput of high-priority can be increased. For easy understanding, we describe the PaSS protocol using the example shown in Fig. 12. For the sake of simplicity, let us assume that only two levels of priority, high and low, are supported by PaSS. Therefore, one bit extension in the packet header (Fig. 8) is enough to represent the priority of the packet. The number of priority levels can be easily adjusted by varying the number of priority bits. The operation of Pass is as follows:

In the scheduling period: The obstacle case is shown in Fig. 12(a). The SC-D is already scheduled by Data1 (low-priority), and Data2 (high-priority) wants to schedule SC-D at a similar time line. In this example, the CM-RTS packet for Data1 has already passed through nodes of (A, B, C, and D). The CM-RTS packet contains the priority of Data1 (i.e., low-priority) because the priority of Data1 is embedded to the CM-RTS packet, as shown in Fig. 8. As node Y is a neighbor node of node D, node Y has overheard the CM-RTS packet. Thus, node Y knows that the priority of Data1 is low. Node Y also knows that the priority of Data2 is higher than that of Data1 because it is the sender. Then, node Y sends a Schedule Switch Request (SSR) to node D and preempts the schedule of Data1, in Step (1). As Data2 preempts the schedule for Data1, there is no schedule for Data1. To set new schedule for Data1, node D sends a RESchedule (RES) packet to node C, in Step (2). By overhearing this RES packet, node Y is acknowledged that the schedule preemption is confirmed. Then, node C sends a CM-RTS to set another schedule for Data1 and the new schedule is made with time gap to avoid collision with the schedule of Data1, in Step (3). Parts of SC-B nodes, which have overheared the CM-RTS, also reschedule the schedule of Data1 to avoid collision with that of Data2. By the new CM-RTS packet, the priority level of the schedule for Data2 is also updated to high-priority, for the preempted schedule by Data2 not to be preempted again by another data packet. In this manner, even though there is a schedule for low-priority packet, the high-priority packet is scheduled ahead. For simple explanation, we set node Y to a data packet sender, but even though a CM-RTS packet is forwarded from
another node to node Y, the schedule is preempted in the same manner.

In the forwarding period: Data packets are forwarded based on the schedule in Fig. 12(b). Data1 from node A stops at the forwarder in SC-B, and waits until Data2 is forwarded through SC-D. After that, the forwarder in SC-B transmits Data1 to SC-C. Each data packet contains the SID which it has to use. Thus, node A transmits Data1 with SID A0, but the forwarder in SC-B changes the SID to A50 and transmits Data1. Then, the high-priority packet is forwarded earlier based on the schedule. Additionally, if Data1 and Data2 are of equal priority, Data2 is forwarded after Data1 is forwarded as in other approaches [3], [4], [12]. In this manner, Pass solves the obstacle in Fig. 12(a) and the throughputs of high-priority packets are increased. Moreover, the overhead of Pass is small.

The overhead for a schedule switch, just two additional packets, a Schedule Switch Request (SSR) and a RE Schedule packet (RES), are additionally needed. As CM-RTS packets pass scores of hops during a scheduling period, the overhead for schedule switching (i.e., 2 hops for SSR and RES) is relatively small. In the simulation, the throughput of high-priority packets are clearly increased by PaSS, but the amount of total throughput degradation by PaSS is just 4%.

Additionally, for the loss of new control packets (e.g., SSR and RES), the retransmission is also used like CM-RS loss. For example, if node C fails to receive the RES packet from node D, there would be no new CM-RTS transmission from node C. In that case, node D retransmits the RES packet after time delay.

IV. EXPERIMENTAL RESULTS

A. Setup

We implemented our GES-MAC as well as previous protocols, namely, LASMAC [3], [4], SMAC [1], BMAC [10], and COR [17] using Tip30 nodes and using NS-2 version 2.33 for performance comparison. Except COR which does not use duty cycling, all other implemented protocols have duty cycle, but the ratio is different. GES-MAC, LAS-MAC, and SMAC have the same 1:4 ratio of a listen period (i.e., scheduling period in GES-MAC) and a cycle, which means the duty cycle is 25%. The lengths of the listen period and cycle were set to 0.5 and 2.0 seconds, respectively. In BMAC, the lengths of the listen period and cycle were set to 2 ms and 200 ms, which means that the duty cycle is 1%. We also implemented a naive combination of COR and SMAC (or RLMAC [9]), so-called SMAC+COR, to examine whether SMAC+COR can achieve our goals (described in Sect. 2) simultaneously. In SMAC+COR, during the listen period, data packets are opportunistically forwarded as in COR, and all nodes sleep during the sleep period.

In order to test the protocols in a multi-hop environment of multiple sources, the experiment should be done in a wide area with many sensor nodes (e.g., one thousand sensor nodes). However, it is very difficult to deploy one thousand sensor nodes. Thus, we implemented the protocols on Tip30 based on MICA2 for the environment of one or two sources.

In tests with Tip30, thirty Tip30 nodes were arranged in 2 m × 40 m as shown in Fig. 6(b). Tip30 nodes are arranged in three rows and ten columns. The row gap is 1 m and the column gap is 4 m. The source is located at the first column and the destination is placed at the tenth column. The number of sources is one or two. The packet dead-line is from one second to five seconds. Tip30 uses CC1000 transceiver and uses 915 MHz, in Fig. 6(a). The other simulation and test parameters are summarized in Table 1.

For a routing protocol, we use a geographic routing [28]. The neighbor node whose Dist Gain (mentioned in Sect. 3.2.1) × Reliability is maximum is selected as the next routing node.

The size of the RTS/CTS/DATA/ACK of each protocol is a little different. In SMAC, RTS/CTS/DATA/ACK/SYNC is 10 bytes/10 bytes/45 bytes/10 bytes/9 bytes each. In COR, the RTS packet piggybacks the IDs of selected candidates. Thus, the size of an RTS packet is 20 bytes. In GES-MAC, data packets piggyback the information about selected next nodes. Data packets also carry a schedule ID. Thus, the size of the data packet is 13 bytes larger than that of other approaches. In BMAC, the preamble packet whose size is 271 bytes is used.

B. Throughput Performance

In the test with Tip30, we had checked this because the feature also influences supporting short packet deadline. We measured the throughput of each protocol by changing the deadline of the packets. Figure 7 shows the
result when there is only one source. If packets whose
deadline are 5 seconds are generated, all protocols except
SMAC provide high throughput. However, the number of
protocols which provides high throughput is reduced as
the packet deadline is shortened. If packets whose
deadline is 3 seconds are generated, the throughput of
BMAC and SMAC + COR become zero. If packets whose
deadline is 1 second are generated, the throughput of
LASMAC becomes low. Not like LASMAC, even in the
case of 1 second dead-line, if the packet is generated
periodically, the throughput of LASMAC becomes low. But unlike LASMAC, even in the case of 1 second dead-line, if the packet is generated periodically, GES-MAC can provide high throughput like
COR. If the packet is generate randomly, the influence by
the initial delay overhead, discussed in Sect. 3.2.5, is big
because of short packet deadline. Thus, the throughput of
GES-MAC is less than that of COR, but GES-MAC still
provides higher throughput than other energy efficient
protocols.

C. Energy Consumption in a Radio Module

We measured the average energy consumed by a
node’s radio module of Tip30 and the result about the
energy consumption of GES-MAC discussed in Sect.
4.2.1 is also measured similarly in the real sensor nodes.
In order to measure the energy consumption of Tip30, we
check the amount of time in idle, transmitting, receiving
state, and calculate the energy consumption with the
parameter in Table 1. One source periodically generates
packets every ten seconds. COR has no duty cycle, 
BMAC has 1:100 duty cycle, and the other protocols
have 1:4 duty cycle, as mentioned in Sect. 4.1. The result
shown in Fig. 9 indicates that the energy consumptions
of the duty cycle-based protocols are much lower than that
of COR because the dominant part of energy
consumption is idle energy. The energy consumption of
GES-MAC is 28% of COR’s energy. GES-MAC consumes 5% more energy than LASMAC on average
because the test environment is less dense than the
simulation environment (in simulation, 15% difference).
The more consumed energy than LASMAC is cost for
providing high throughput. Like simulation, BMAC is
very energy efficient although it has difficulties in
providing high throughput.

D. Discussion about the Overhead of GES-MAC

We need to discuss the overhead of GES-MAC to
provide energy efficiency while maintaining high
throughput. First, we discuss the number of control
packets. We measured the number of control packets with
Tip30 nodes. Figure 8 shows the number of detected
control packets at a node in the center area. The control
packets are RTS/CTS/ACK/SYNC/Preamble. Compare
to COR and BMAC, GES-MAC needs more control
packets. As GES-MAC uses synchronized duty cycle like
LASMAC and SMAC, at the beginning of every listen
periods, SYNC packets are shared even though there are
no data packets to forward. We think that the control
packet overhead is a cost of GES-MAC in order to
balance high throughput and energy efficiency.

Additionally, because BMAC needs a preamble packet
for data forwarding, BMAC needs more control packets
than COR. Interesting feature is that COR needs the
fewest number of control packets, but consumes very
large energy.
After analyzing the compiled size of code, we thought that the code complexity is not proportional to the size of compiled code because the size of code can be changed by programmer’s skill. For example, in our version, the code size of SMAC is 50 kbytes and that of LASMAC is 46 Kbytes. However, theoretically, LASMAC is more complex than SMAC because LASMAC needs node scheduling. Thus, with the code that we have, we cannot address the code complexity. Just note that the code size of GES-MAC is 56 Kbytes.

Note that we set the size of sub-slot in Sect. 3.2.3 to two time slots (i.e. 2 ms). The computing power of Tip30 is not so good and each sensor has different computing power. Thus, if we set the sub-slot to 1 ms, two nodes try to send ACK simultaneously and collisions are induced. However, after changing the size of sub-slot to 2 ms, GES-MAC operates as we expected.

E. Experimental Results for Service Differentiation

We measured the service differentiability of GES-MAC, which is the effects of the Priority-aware Schedule Switching (Pass) on the throughput of high-priority packets, with two close Tip30 sources. One source is a high-priority source (source1) and the other is a low-priority one (source2). In this test, the previous service differentiation using different Contention Window (CW) is applied. Source1 has 16 slots for CW and Source2 has 32 slots for CW. The time deadline is 1 second. Without PaSS, if a source2 schedules SCs through the routing path earlier, the packet from source1 waits for long time gap and the packet does not reach the destination within one second. Thus, 250 high priority packets meet the deadline. However, if Pass is applied, even though SCs are already scheduled for source2, the high-priority packets are forwarded earlier by schedule switching. As a result, 360 high-priority packets (i.e. all populated packets) meet the deadline. The increased high-priority throughput means the good service differentiability of Pass.

V. CONCLUSION

The goals of our work are to provide high throughput in duty-cycled environments and to provide service differentiation while maintaining high throughput. We proposed a method, called GES-MAC, which satisfies the goals. GES-MAC reduces idle listening energy consumption by using a synchronized duty cycle. Cluster-based multi-hop scheduling pro-vides high throughput in duty cycle environments. Priority-aware schedule switching makes more high-priority packets reach the sink node. Experiments showed that GES-MAC reduced the idle listening energy consumption by about 70% by using a duty cycle. The throughput of GES-MAC is not much less than that of COR. GES-MAC also provides service differentiation with little overhead. Therefore, GES-MAC achieves our goals.

REFERENCES


Jianhua Wang was born in 1956. Professor and Ph. D. supervisor in Harbin Normal University. Her main research interests include image processing, intelligent information processing.

Yan Yu born in 1976-02-23, Harbin, Heilongjiang province, China. He is currently a Associate Professor in Harbin Normal University. His research interests include knowledge engineering, information retrieval, intelligent education.

Madukasi Charles Nnaemeka is Nigerian, born in 1989. He has got degree from school of international exchange, Harbin Normal University. His main research interests include image processing and artificial intelligence.