MAC Support for Wireless Multimedia Sensor Networks

Osama Farrag*, Mohamed Younis**, and William D'Amico*

*Milton Eisenhower Research Center, Johns Hopkins Applied Physics Lab., Laurel, Maryland, Osama.Farrag@jhuapl.edu
**Dept. of Computer Science and Elec. Eng., University of Maryland Baltimore County, Baltimore. Maryland, younis@cs.umbc.edu

Abstract— A wide range of applications such as disaster management, military, and security have fueled interest in sensor networks in recent years. Since the employed sensor nodes often are significantly constrained in their onboard energy, most of the research work, including that on medium access control (MAC) protocols, has focused on optimizing energy consumptions and cared less about data delivery latency. However, employing advanced video and audio sensing devices needs special attention to quality of service requirements associated with multimedia data. This paper presents MAQ, a novel Medium Arbitration scheme for supporting QoS traffic over a single communication channel. MAQ decentralizes the control of the communication resources and fully integrates TDMA and CSMA/CA schemes for medium access. The main objective of MAQ is to ensure predictable delay and fair access for the nodes while achieving high channel utilization and network throughput. The simulation experiments confirm the effectiveness of MAQ.

I. INTRODUCTION

Recent years have witnessed an increased interest in applications of wireless sensor networks (WSNs). Examples of these applications include search-and-rescue, border protection, and battlefield surveillance, where WSNs are deployed in remote and inhospitable environments to collect data without human risk. It is envisioned that WSNs will consist of numerous miniaturized sensor nodes that report their findings to a base-station. WSNs are expected to be formed in the field in an ad-hoc manner. Due to the lack of infrastructure, the remote deployment, and the uncertainty about the inter-node communication links, a WSN is usually bootstrapped and managed in a decentralized manner.

The limitation of sensor energy has motivated substantial research on WSNs. The bulk of the efforts have been dedicated to minimizing the energy consumed in communication. The network and link layers have received the most notable attention. At the network layer, the focus has been on energy-aware route setup [1]. The main idea is to minimize the transmission power, which is proportional to distance squared, through the pursuance of multi-hop data forwarding so that the cumulative transmission energy is reduced compared to direct sensor to base-station interactions. Energy-efficient MAC protocols tackle the energy wastage due to collisions among the radio transmission of nodes, keeping the receiver unnecessarily active and the excessive state changes of the radio circuit [2].

While optimizing energy consumption is a plausible objective, meeting quality of service (QoS) requirements is still necessary. This is particularly important for applications of multimedia sensors, which involve delay sensitive traffic. In fact these applications are getting more popular lately and satisfying QoS requirements is becoming a prominent design goal [3]. However, little attention has been paid to dealing with QoS traffic in the context of WSNs, mostly at the network layer. Very few attempts have been made to devise QoS MAC protocols to support multimedia WSN applications.

This paper presents a novel Medium Arbitration scheme for supporting QoS traffic (MAQ) in Wireless Multimedia Sensor Networks (WMSNs). MAQ is geared for providing predictable medium access delay at the level of the individual nodes, without sacrificing the utilization of the available bandwidth. To achieve that over a single channel, MAQ fully integrates TDMA and CSMA/CA medium arbitration schemes. A node is allowed to schedule its transmission for an entire session and guarantee collision free access until the session ends. Contention based access is also supported in order to boost the throughput and resource utilization when the entire network needs to support a mix of synchronous, constant bit-rate, and asynchronous, variable bit-rate, burst traffic. Unlike a contemporary implementation of CSMA/CA, MAQ interleaves contention-free and contention-based periods in order to spread the demand and minimize collisions. The predictable medium access not only is invaluable for supporting QoS traffic but also allows nodes to employ energy conservation measures such as switching to low-power sleep mode. Such capabilities enable scalability and robustness. To the best of our knowledge no published protocol provides all these features. The simulation results confirm the effectiveness of MAQ.

The paper is organized as follows. The next section describes the system model and highlights the design goals and philosophy of MAQ. Section III contrasts MAQ with related schemes in the literature. Section IV provides a detailed description of MAQ. In section V, the performance of MAQ is studied through simulation. Section VI concludes the paper.

II. PROBLEM MODEL AND SOLUTION RATIONALE

A. System Model and Design Goals

The considered system involves a set of stationary multimedia, e.g. video, sensors that are spread throughout an area of interest. A sensor is equipped with a limited data processing engine and a radio. Sensors are only capable of short-haul communication and are responsible for probing the environment to detect a target/event. The radio onboard a sensor node is capable of transmitting and receiving over a single frequency channel. Each sensor is assigned a unique ID prior to deployment. A sensor is assumed to be capable of operating in active or low-power stand-by mode.

Data, e.g., video captures, collected by sensors are disseminated over multi-hop paths to a base-station that is located in the field. The base-station fuses the sensors’ data and interfaces the network with a remote command center via a high-speed long-haul communication link. The delivery of sensor data to the base-station is subject to latency constraints. Some sensors will stream their data continually while others report only when detecting something of interest. The former constitutes periodic real-time data streams while the latter
creates sporadic delay-sensitive sessions. Dealing with these latency concerns often involves special consideration at multiple layers in the protocol stack, especially the network and link layers. This paper focuses only on MAC support.

Generally, an efficient MAC layer protocol for WSNs should have the following design attributes:

- The protocol should allow predictable and deterministic access time in order to meet latency constraints on the delivery of data often generated by multimedia sensors.
- Collisions among the transmissions of various nodes should be minimized. Collisions lead to packet drop and thus reduce throughput and cause energy wastage.
- The protocol should be scalable since most applications of WSNs involve numerous sensor nodes.
- Given the resource constraints, the protocol should impose low overhead and enable energy conservation measures such as switching to a low-power sleep mode in order to minimize energy wastage in idle listening.

B. Solution Rationale

Due to the large node population in WSNs, CSMA/CA has been viewed as the most scalable scheme for such networks by many researchers. However, a number of issues impact the efficiency of CSMA/CA in multi-hop topologies and present the potential of very large medium access delay [4]. The latter concern is particularly serious for WMSN since the transmitted data is often subject to time constraints. To accommodate latency sensitive traffic, access delay should be both predictable and bounded. One way to achieve this is to ensure exclusive access to the medium when a node \( S_i \) has to transmit time-constrained data. The exclusive access can be implemented by partitioning the available bandwidth into disjoint frequency bands and assigning a distinct unshared band to \( S_i \). However, such frequency division often yields low bandwidth utilization and does not scale for large networks. The alternative approach is to pursue time-based arbitration and provide for \( S_i \) a reserved schedule with exclusive access to the medium. While temporal partitioning can prevent collisions and increase the throughput, it is inflexible when static allocation is enforced, leads to resource wastage if all nodes get guaranteed a share regardless independent of need, and requires high overhead when implemented in a distributed manner.

To support latency sensitive traffic while achieving both scalability for large networks and good bandwidth utilization, MAQ provisions contention free and contention based access to the medium. In other words, MAQ fully integrates TDMA and CSMA/CA. Contemporary approaches found in the literature that pursue a similar strategy split the frame into a reservation sub-frame within which nodes contend to reserve a medium within a data sub-frame. Unlike these approaches, MAQ allows the interleaving of contention free and contention based periods and thus can enable rapid access to the medium.

III. Related Work

MAC protocols proposed for WSNs can be classified into centralized, e.g. [5], with the base station scheduling medium access and distributed, such as [6], where nodes coordinate among themselves. Some employ a single channel [5][6]; others use multiple channels [7]. While some apply contention-based schemes [7][8]; others use reservation-based ones [5][6]. Most published MAC protocols for WSNs optimize power consumption with little attention paid to QoS constraints [2]. Given the design of MAQ, the focus in the rest of this section will be on MAC protocols that employ hybrid medium sharing methodologies or attempt to support QoS goals.

SMACS employs a hybrid TDMA and FDMA medium sharing scheme [6]. A flat network topology is assumed. Sensors agree with their neighbors on a schedule for medium access. Interference between adjacent links is avoided by randomly assigning different channels to potentially interfering links using FDMA. MAQ uses only a single channel to achieve better utilization of the available bandwidth. In addition, frames in SMACS tend to be significantly large increasing latency. Meanwhile, I-EDF [9] assumes a grid architecture. Nodes at the border of a cell are assumed to have two radio transceivers that are tuned to distinct channels. Nodes in a cell exchange their sampling frequency and each applies an earliest deadline first scheduling strategy in order to avoid medium access collision and meet timing constraints. MAQ works on a single channel and does not assume a grid structure.

On the other hand, PARMAC considers a tiered network topology [10]. Intra-cluster communication is based on time-based medium sharing. Inter-cluster collisions are avoided by assigning distinct slots to every cluster. Others, like [5] and [11], try to address the potential of inter-cluster collisions by changing the slots assigned to conflicting transmissions in neighboring clusters. However, these approaches often yield large frame sizes and data latency. In addition, the two CHs must be directly reachable to each other to coordinate slot assignment and reassignment when needed. MAQ efficiently addresses the inter-cluster interference by allowing a gateway node to make a reservation in a contention free period that fits the transmission schedule of multiple clusters. In other words, conflict avoidance solution is simply applied by the gateway node without requiring high level coordination. In addition, contention based access is not allowed in other approaches making them inflexible and resource inefficient.

Almost all efforts on supporting QoS at the MAC layer have been dedicated to providing service differentiation for the various traffic types. Most of these schemes slice a frame into two parts; a contention period to make a reservation and contention free periods for sending data. For example, Q-MAC assigns priority to traffic based on the latency requirements [12]. MACAW, which is popularly used in wireless networks, is the basic underlying medium access protocol. The authors extend MACAW to support selective sleep mode by introducing time slots in the data period that can be reserved in the contention period, and allowing the node to know when to be active. A similar idea is pursued in [13]. However, the authors further introduce an interactive period for probing the environment and turn off their radios. MAQ interleaves contention-based and contention-free periods and thus allows low latency channel access. Moreover, these approaches in general assume per-packet reservation and tend to boost the delay jitter, which should be bounded in MWSNs. Some other schemes pursue CSMA/CA and adaptively set the contention window based on the traffic priority [14] or volume [15]. Obviously, medium access stays mostly non-deterministic.

IV. Medium Arbitration for Supporting QoS Traffic

MAQ fully integrates TDMA and CSMA/CA schemes to support fair and predictable medium access for delay sensitive traffic. The MAQ design strikes a balance between centralized and distributed medium access reservation methodologies. On
the one hand, centralized approaches do not scale and have a single point of failure. On the other hand, a totally distributed approach has a high coordination overhead or does not fully eliminate the potential for collisions. MAQ pursues a decentralized approach where medium control is split among multiple nodes and decisions about medium access are localized. The underlying architecture for MAQ involves partitioning the network into clusters. Medium access within a cluster is autonomous. Inter-cluster interference is handled at the level of the gateway nodes in the individual clusters.

A. 2-hop Clustering

The operation of MAQ is based on a 2-hop clustering of the WSN. Sensors group themselves into overlapped clusters by applying a distributed, randomized clustering algorithm such as MOCA [16]. The goal of the clustering process is to ensure that each node is either a cluster head (CH) or within 1 hop from at least one of the CHs. In other words, nodes in the cluster can be as far as 4 hops away from each other and the path between any pair of CHs have at least two hops in between. Basically, when a node is powered up, it tries to hear a beacon from any existing CH. If no beacons are heard for some predetermined duration, the node elects itself as CH and transmits a beacon to announce the cluster to its neighbors. Directly reachable nodes join the self-acclaimed leader and also participate in the cluster management activities by transmitting beacons to inform their neighbors that could be 2-hops away from the CH that they are within an established cluster coverage area. Details about how long a node waits before nominating itself as a CH can be found in [16].

In MAQ, controlling medium access is not exclusive for the CH. Instead, a group of nodes in the cluster are charged with that task. This group is referred to hereafter as Cluster Control Group or CCG for short. Clearly, this decentralized approach avoids making the CH a bottleneck and a single point of failure for the cluster. The CCG is composed of the CH and a subset of its directly reachable nodes, i.e., 1-hop neighbors. These nodes will take turns in transmitting beacon messages at fixed intervals. A beacon message marks the beginning of a frame so that the nodes synchronize their clocks. In addition, the message includes a medium access schedule to all nodes in the cluster in order to prevent collisions during reserved time periods within the frame. Thus, the beaconing overhead, e.g. transmission energy or bandwidth, per CCG member is:

\[
\text{Beacon overhead} = \frac{\text{Cost of beacon message}}{\text{Frame size} \times |\text{CCG}|} \quad \text{(1)}
\]

Since the medium access control duties are shared among the CCG nodes, synchronizing the cluster state, i.e., medium access schedule, at these nodes is necessary. As detailed later, a state update message is implicitly sent when it is the turn of a CCG member to broadcast a beacon. In order to avoid having inconsistent state among the CCG members for long, the size of the CCG set is constrained to a preset number depending on the data access latency that the application can tolerate. The relationship will be clear when the details of the reservation are discussed. Therefore, some admission policy is enforced at the time of cluster formation. Basically, 1-hop neighbors of the CH do not become by default part of the CCG when they decide to join the cluster. Instead, the CH selects up to the maximum allowable number. Obviously, if only few neighbors join the cluster, they all become members of the CCG. Nodes which are not able to hear beacons generated by any CCG member will not join the cluster and either join other clusters or form adjacent clusters on their own. It is worth noting that a sensor that becomes part of multiple clusters serves as a gateway node. Fig. 1 shows an example of 2-hop clustered network architecture, highlighting the role of the various nodes.

The obvious question is why to pursue a 2-hop clustering strategy if arbitracting access to the medium in a 1-hop cluster can be simply performed by the CH. While it is indeed easier to do so, a number of drawbacks exist and warrant a larger cluster radius than 1-hop. First, the CH becomes a single point of failure if it is to handle all access requests. Second, having 2-hop clustering reduces inter-cluster coordination, since it yields fewer clusters with larger non-overlapping regions. However, going beyond two hops increases the management overhead as explained below. It is worth noting that since the nodes are stationary, cluster maintenance will be needed only if a node fails. The reaction depends on the role of the failed node. For a failure of a regular node it suffices for the closest CCG member to announce that in the next beacon message and any CFPs allocated to that failed node are reclaimed. A failure of a CCG member will involve more action. Basically the CH has to make sure that all leaf nodes in the cluster that are 2-hop away from the CH are covered. The CCG set may have to be augmented with additional members to replace the failed one. Finally, if the CH fails, the cluster has to split. The CCG members will run the clustering algorithm again as explained above. Two or multiple clusters may result. The impact on neighboring clusters will be limited only to the gateway nodes.

B. Medium Access Control

MAQ enables the nodes within a cluster to collaborate in partitioning the capacity of a common wireless channel into non contiguous Contention Free Periods (CFPs) and Contention Periods (CPs). Each CFP is allocated to a specific node for exclusive access, while all nodes share the available CPs using CSMA/CA. In essence, MAQ provides simultaneous support for TDMA and CSMA/CA over a single channel to permit efficient communication and to support time-sensitive data in WMSNs. The frame structure is shown in Fig. 2.

![Fig. 1: A clustered network configuration with a cluster radius of 2.](image1)

![Fig. 2: Contention-based and contention-free periods are interleaved in a MAQ frame and beacons marks the frame boundaries.](image2)
Basically, beacons are transmitted at the beginning of every frame to synchronize the nodes’ clocks in the cluster. The following describes the various functions of MAQ.

1) Resource Management: The responsibility of arbitrating medium access among the nodes in a cluster is shared by the CCG members. In other words, the CH does not solely bear the management load of the cluster. The duties of the CCG are two fold. First, they are to transmit beacons in order to synchronize the clock of the nodes in the cluster. Basically, the CCG members take turns sending the beacons. In addition to the order in which CCG members transmit beacons, a beacon also includes the schedule of CPs and CFPs in the frame. Since not all nodes in the cluster are reachable to the CH or its 1-hop neighbors, rotating the responsibilities among the CCG members causes some nodes to miss the beacon. To tackle this problem, MAQ introduces contended beacons in a designated period, as seen in Fig. 2. If it is not the turn for a certain CCG member $CCGi$ to transmit the beacon, and $CCGi$ was not able to hear any beacon during the scheduled beacon period, then $CCGi$ steps forward and contends to transmit a beacon unless another CCG within its range was able to transmit a beacon before it. These contended beacons are required in order to inform the nodes that cannot hear the beacon that a scheduled beacon is not heard. $CCGi$’s decision to transmit a beacon is based on the proximity of its neighbors to the CCG member that is scheduled for sending the beacon in the current frame. Contended beacons also allow recovery from a sudden failure of a CCG member. The cluster nodes adjust their clocks when they hear the beacon marking the beginning of a new frame. MAQ does not force clock synchronization across clusters. This makes MAQ very scalable. A gateway node will have to align the distinct beacons that it receives from different clusters. Fig. 3 illustrates the idea.

The second duty of CCG members is to maintain the medium access schedule. Reservations have to be made through a CCG member. Two important issues arise with such a decentralized reservation management strategy. The first is how to ensure a consistent view of the current cluster state in order to generate a conflict free medium access schedule. In other words, the CCG members should be aware of all reservation requests even if a request is received and handled by only one member. This is a classical state synchronization issue faced in all distributed systems and can simply be addressed by frequent exchange of update messages. However, such solution would introduce prohibitive overhead, particularly because the CCG members are not directly connected. Recall that the CCG is composed of the CH and its 1-hop neighbors. In addition, the reservation and the response are not guaranteed to be heard by all CCG members to keep the state updated. In MAQ, the beacons are used to detect inconsistency and triggers fixes. However, there is still a possibility that the beacon is not heard and that the update is not made. MAQ addresses this issue by constraining the reservation approval as explained next.

2) Reservation and release: Unlike most popular reservation based medium access protocols, MAQ does not centralize the time slot reservation process. Instead a node makes a reservation of a context free period by contacting one of the members of the CCG. One of the distinct features of MAQ is that reservation, if approved, will last for the entire session and will not have to be made for every frame. In other words, a successful reservation schedules a CFP for the requesting node for every frame until the requester explicitly demands a release. This feature is invaluable for multimedia traffic such as video streams that are continuously transmitted and subject to latency constraints. In order to avoid bandwidth loss if a node fails while having unreleased CFP, the closest CCG member monitors the usage and de-allocates the CFP if it is not used for a number of consecutive frames. Gateway nodes will have to make a reservation in all of its adjoining clusters with synchronized start of CFP in all these clusters. Thus, no explicit inter-clusters synchronization is performed.

For a reservation to be successful, a distinct CFP needs to appear in the beacon. Thus, the requesting node will check the beacon of the next few frames in order to know the results of the reservation. Assume that a node $nj$ submitted a reservation request to $CCGi$. $CCGi$ will then check the current medium access schedule and decide whether allocating a CFP is possible. However, $CCGi$ cannot commit the reservation without ensuring the consistency of its view of the cluster state with that maintained at the other CCG members. MAQ does not designate the specific time in the frame for exchanging the state update to avoid imposing overhead. The rationale is that most reservations will be made upfront and will last for a while, and thus the frequency of reservation requests is not high after the first few frames. In addition, the state update comes for free when $CCGi$ hears a beacon from another CCG member. Recall that the CH is reachable to all other CCG members and it is thus guaranteed that at least one beacon can be received by another CCG member. Therefore, $CCGi$ will wait for its turn to send the beacon before committing the scheduled CFP. The requester will wait until that moment to know the result of its reservation. Fig. 4 shows a state diagram model for the operation of non-CCG and CCG nodes according to MAQ. Mathematically, the worst-case reservation time is:

$$\text{Reservation time} = \lceil |\text{CCG}| \times \text{Frame Size} \rceil \quad \text{(2)}$$

Although it may take time for a request to be granted, a reserving node can use CPs in the meanwhile. In addition, the request is often for a long lasting session, and thus any medium access delay suffered will be transitional until a CFP is scheduled. It is important to note the effect of the carnality of the CCG set. While having many members allows splitting the management load on more nodes, a reservation request will take longer since the rotation cycle among the CCG members becomes larger as also indicated by (2). Obviously, this issue is subject to a trade-off and is also affected by the tolerable reservation delay in the context of the particular WMSN application and by the frame size, as captured by (3).
This metric is used to the handling of request and participation in the beacon State machine representation for the protocol regarding (a) Fig. 4:

C. lightweight heuristics would suffice. If CH are eligible, the scope and complexity are limited and most NP-hard [17]. Nonetheless, given that only the neighbors of the members is a classical set cover problem that is known to be set by the CH based on (3). Picking the minimal CCG cluster connectivity. Additional neighbors are augmented to the minimal set of 1-hop neighbors required for strong intra-cluster interaction. Consequently, the CCG members are essential for reaching the node. To cope with any application-imposed latency constraint on reservation response time, either fewer CCG members are allowed or the frame size is reduced. Generally, the frame size is dependent on the clock drift rate. The clock drift rate will put a bound on how big the frame can be. Having smaller frames than required for maintaining clock synchrony would increase the beacon frequency and boost the overhead. On the other hand reducing the size of the CCG set can also be constrained by the fact that certain CCG members are essential for reaching certain nodes in the cluster. Therefore, in MAQ the CH picks the minimal set of 1-hop neighbors required for strong intra-cluster connectivity. Additional neighbors are augmented to the set by the CH based on (3). Picking the minimal CCG members is a classical set cover problem that is known to be NP-hard [17]. Nonetheless, given that only the neighbors of the CH are eligible, the scope and complexity are limited and most lightweight heuristics would suffice.

C. Energy Conservation Measures

Since MAQ enables predictable medium access, many of the contemporary energy measures found in the literature can be employed. Clearly, a node that knows when to transmit does not have to contend for medium access. Thus, medium collision is mostly avoided in MAQ, not only at the intra-cluster level but on the inter-cluster level as well. MAQ offers two provisions that support selective switching to sleep mode at the receiver. First, the reservation can include the ID of the receiver, implicitly informing other nodes that they are not involved and turn to low power mode. Nonetheless, exploiting this feature would require cross-layer optimization through interaction with the network layer. Basically, the route should be fixed or at least stable within the cluster so that the reservation stays valid. Second, a node can wake-up at the beginning of a CFP and decide to sleep if there is RTS for it. This applies for CFP if the receiver is not known. It is worth noting that MAQ can also be extended to broadcast the active/sleep schedule of the nodes in the cluster.

Energy can also be saved at the level of CCG members. First, sharing the role and rotating the beacon broadcast among the CCG members balances the load and allows them to conserve some of their resources. Second, if sufficient coverage is ensured, some CCG members can switch to sleep mode for one or multiple frames. Recall that the CH employs a set cover heuristic to pick the minimal count of the CCG nodes. The CH can pick redundant CCG nodes, as long as the maximum the CCG count is not reached, as explained in Section B above. Since the engagement schedule for a CCG member is predictable, a CCG, like other nodes in the cluster can take advantage of a CFP and switch to a low power mode.

V. PERFORMANCE EVALUATION

A. Experiment Setup and Performance Metrics

MAQ is validated using the ns2 simulator. The goal of the setup reported in this section is to enable a focused validation of some key medium access arbitration features of MAQ. Basically, deterministic placement of nodes is pursued to form a topology with predictable routes to prevent cross-layer interaction issues from affecting the observed measures at the MAC layer. Nodes are placed on a two parallel line segments that are 100 meters apart. Half of the N deployed nodes are positioned on each segment with 1 meter distance apart. The node’s communication range is set to 250 meters. Fig. 5 shows an illustration. With values of N less than 500, all nodes stay in the range of each other and the potential of collisions becomes very high. This experiment setup allows validating the MAQ performance under stretch. Nodes are equipped with omni directional antennas and a two-way ground propagation model with a channel rate of 1 Mbps. The frame size for MAQ is set to 3 sec. Full duplex UDP sessions are established between the pair of corresponding nodes on the two line segments, i.e. node i communicates with node N/2+i. Traffic is generated by each node at a constant 90K bit/sec with a packet size is 2000 Bytes. MAQ’s performance is being studied under numerous metrics. However, due to space constraints, we report only on the following subset, which captures the scalability, resource efficiency, fairness and timeliness properties of MAQ:

- **Achievable Channel Capacity:** This metric is used to measure the level of conflicts (and eventually interference) in medium access. It is calculated as the total number of data bits transmitted in a frame. MAQ is expected to utilize most of the available bandwidth in data transmission.
- **Delay Jitter:** This metric is very important for multimedia data and is of great interest in WMSNs. It captures how MAQ can achieve a predictable delay and enable all nodes to have their fair share in the medium.

B. Simulation Results

In each experiment, the number of sensors has been varied. Each node is assigned a CFP with 1ms duration. The reported results are on individual simulation runs, each lasting for 1500 sec. The IEEE 802.11 is used a baseline for comparison.
This paper has presented a novel Medium Arbitration scheme for supporting QoS traffic (MAQ) in WMSNs. MAQ fully integrates TDMA and CSMA/CS over a single channel in order to provide predictable medium access latency and to support delay-sensitive traffic. MAQ pursues a decentralized management strategy in which multiple nodes share the duties of arbitrating medium access among the nodes and synchronizing their clocks. The validation results have demonstrated that even with high sensor densities, MAQ achieves great utilization of the available bandwidth and provides the application packets with a close to zero delay jitter. Such performance is invaluable for multimedia traffic. The observed maximum and average delay jitter have been very similar indicating that MAQ enables fair medium access among the nodes. In addition, the simulation experiments have demonstrated that MAQ scales very well for large networks. MAQ provides numerous opportunities for cross layer optimization, a feature that we intend to exploit in the future for fine-grained performance tuning.

VI. CONCLUSION

This paper presented a novel Medium Arbitration scheme for supporting QoS traffic (MAQ) in WMSNs. MAQ fully integrates TDMA and CSMA/CS over a single channel in order to provide predictable medium access latency and to support delay-sensitive traffic. MAQ pursues a decentralized management strategy in which multiple nodes share the duties of arbitrating medium access among the nodes and synchronizing their clocks. The validation results have demonstrated that even with high sensor densities, MAQ achieves great utilization of the available bandwidth and provides the application packets with a close to zero delay jitter. Such performance is invaluable for multimedia traffic. The observed maximum and average delay jitter have been very similar indicating that MAQ enables fair medium access among the nodes. In addition, the simulation experiments have demonstrated that MAQ scales very well for large networks. MAQ provides numerous opportunities for cross layer optimization, a feature that we intend to exploit in the future for fine-grained performance tuning.

REFERENCES