Ann. Telecommun. (2009) 64:705–721
DOI 10.1007/s12243-009-0095-0

Aggregated traffic flow weight controlled hierarchical MAC protocol for wireless sensor networks

M. Abdur Razzaque · M. Mamun-Or-Rashid · Muhammad Mahbub Alam · Choong Seon Hong

Received: 1 September 2008 / Accepted: 20 March 2009 / Published online: 1 May 2009

Abstract It has been discussed in the literature that the medium-access control (MAC) protocols, which schedule periodic sleep–active states of sensor nodes, can increase the longevity of sensor networks. However, these protocols suffer from very low end-to-end throughput and increased end-to-end packet delay. How to design an energy-efficient MAC protocol that greatly minimizes the packet delay while maximizing the achievable data delivery rate, however, remains unanswered. In this paper, motivated by the many-to-one multihop traffic pattern of sensor networks and the heterogeneity in required data packet rates of different events, we propose an *aggregated traffic flow weight* controlled hierarchical MAC protocol (ATW-HMAC). We find that ATW-HMAC significantly decreases the packet losses due to collisions and buffer drops (i.e., mitigates the congestion), which helps to improve network throughput, energy efficiency, and end-to-end packet delay. ATW-HMAC is designed to work with both single-path and multipath routing. Our analytical analysis shows that ATW-HMAC provides weighted fair rate allocation and energy efficiency. The results of our extensive simulation, done in *ns-2.30*, show that ATW-HMAC outperforms S-MAC; traffic-adaptive medium access; and SC-HMAC.

M. A. Razzaque · M. Mamun-Or-Rashid · M. M. Alam · C. S. Hong (✉)
Department of Computer Engineering,
Kyung Hee University, 1, Seocheon-ri, Giheung-eup,
Yongin-si, Gyeonggi-do, 446-701, South Korea
e-mail: cshong@khu.ac.kr

M. A. Razzaque
e-mail: m_a_razzaque@yahoo.com

Keywords Wireless sensor network · Medium-access control · Weighted fair rate allocation · Multipath routing · Energy efficiency

1 Introduction

Recent advancements in the design of sensor node components have enabled the deployment of wireless sensor networks (WSN) capable of supporting a number of applications, ranging from reporting very low data rate periodic traffic to comparatively high bandwidth hungry traffic due to sudden important events or multimedia data delivery [1, 2]. Each application may produce different classes of traffic and impose a unique set of goals and requirements based on the sensed event(s). For example, in forest-monitoring applications, fire detection events produce more critical traffic flows (i.e., demanding high end-to-end throughput and reduced end-to-end packet delay) than events like thunderstorms, cyclones, etc. Similarly, data traffic carrying moving object tracking (e.g., movement of wild animals in a forest) information should get higher weight than its counterpart (e.g., static or slowly moving animals). The traffic heterogeneity in application-specific requirements also appears in other emerging WSN applications like industrial process control, radioactive radiation and leakage detections, vehicle monitoring, etc. Moreover, energy efficiency is one of the most important design criteria for all WSN applications. The underlying medium-access control (MAC) protocol of the network has a significant role in meeting the above demands. To this extent, keeping aside the contribution of upper-layer protocols (i.e., routing or rate control),

in this paper, we concentrate only on the functions of MAC layer.

A large number of MAC protocols for sensor networks exist in the literature [3–9]. They can mainly be divided into two groups: contention-based and (TDMA-like) schedule-based MAC protocols. The contention-based protocols [5–8] use 802.11 DCF for medium access and a variety of periodic sleep–listen schedules to conserve energy. However, due to the intrinsic many-to-one multihop and convergent traffic patterns of sensor networks, the use of 802.11 DCF causes a huge amount of packet loss, as described in Section 3 in detail. On the other hand, in schedule-based MAC protocols [3, 4, 9, 10], nodes exchange traffic and transmit schedule information in order to predetermine and reserve the time slots of their future transmissions. These protocols guarantee collision-free transmission slots. However, for large-scale multihop wireless networks, schedule-based protocols exhibit inherently higher delivery delays when compared to contention-based approaches [3, 9]. Therefore, in today's emerging sensor networks, there is a growing need for improved MAC protocol that can provide higher throughput, reduced latency, and energy efficiency.

In SC-HMAC [11], we first introduced the concept of a hierarchical MAC protocol for WSNs, which was based on the *source count* values of the nodes. SC-HMAC was designed for homogeneous traffic applications and single-path routing networks only. However, in this paper, we design a more generous MAC protocol that adapts to the application at hand and works in networks either with single-path or multipath routing, namely, *aggregated traffic flow weight* controlled hierarchical MAC protocol (ATW-HMAC). ATW-HMAC uses traffic-flow-based information to determine the frequency of medium access of a node (which is determined by the minimum contention window, CW_{min}, value of the node). Nodes using ATW-HMAC calculate their aggregated traffic flow weights (defined in Section 4.2) and thereby determine their CW_{min} values.

A number of other works exist in the literature that adjust the CW_{min} value of nodes to increase the throughput of IEEE 802.11 DCF [12, 13]. In [12], the backoff procedure of DCF is modified to maximize the saturation throughput of wireless LAN stations. The authors of [13] opened up two approaches in order to control IEEE 802.11e station's airtime usage: by controlling $AIFS$ time or CW_{min} and CW_{max} values. Unlike these works, our proposal exploits the event traffic rate heterogeneity and many-to-one convergent traffic pattern of WSN in differentiating the CW_{min} values of nodes. Our work answers the following key questions:

- How can a sensor node, independently and in a distributed manner, determine its appropriate CW_{min} value on the fly?
- Which parameter (*traffic flow weight* or *source count value*) is more effective in determining the CW_{min} values of WSN nodes for performance improvements in terms of throughput and delay?
- How can weighted fair data delivery from multiple events having differences in data packet generation rates be ensured?

Finally, the main contributions of ATW-HMAC are summarized as follows: (1) the distributed maintenance of energy-efficient, collision-avoided MAC based on implicit single-hop traffic information; (2) higher throughput and lower end-to-end packet delay; (3) localized and simple protocol, so that it can be run by nodes with limited processing, memory, and power capabilities; (4) capability to work with various applications and both single- and multipath routing; and (5) weighted fair rate allocation and energy efficiency. The performance of ATW-HMAC is compared with that of S-MAC [6]; traffic-adaptive medium access (TRAMA) [3]; and SC-HMAC [11]. The rest of the paper is organized as follows: Related works are discussed in Section 2, and Section 3 discusses the impact of using IEEE 802.11 DCF in sensor networks. The proposed ATW-HMAC protocol is presented in Section 4, which includes its network and traffic models and determination procedure of nodes' CW_{min} values. Section 5 presents the probabilistic analysis on weighted fair rate allocation and energy consumption made by ATW-HMAC. Performance evaluation is carried out in Section 6, and the paper concludes in Section 7.

2 Related works

A number of MAC protocols exist for WSN aiming to adapt with its traffic behavior in order to reduce the packet drops and to increase the energy efficiency. Some of them are detailed as follows: TRAMA protocol [3] tries to maximize the length of sleep periods by determining the presence or absence of traffic. It needs a neighbor protocol and a medium-access schedule exchange protocol that has significant overhead of broadcasting schedule packets periodically. It also performs an adaptive election algorithm to reduce the number of unused time slots. The key problem of this protocol is that the overall signaling overhead of the

above fairly complicated algorithms may create a scalability problem and increase end-to-end packet delivery latency, particularly for large-scale sensor networks. It also suffers from computation and distribution of global network-wide scheduling information and time synchronization.

Later, the complexity and scheduling overhead of TRAMA are criticized by the same authors in [9], and a new TDMA-based MAC protocol is proposed, namely flow-aware medium access (FLAMA). Unlike TRAMA, FLAMA organizes the time in periods of random and scheduled access intervals. However, the implementation of FLAMA needs the exchange of two-hop neighborhood information, execution of distributed election algorithm at every node, and propagation of schedule information to ensure conflict-free slot assignments. These communications and computations waste not only the sensor node energy but also the transmission slots, which in turn increase the packet delivery delay.

The funneling MAC [4] is a sink-oriented hybrid MAC protocol that uses both schedule-based TDMA and contention-based CSMA/CA MAC schemes. Pure CSMA/CA operates network-wide in addition to acting as a component of the funneling MAC that operates in the high-traffic region. This protocol mitigates the funneling effect by using local TDMA scheduling for the nodes closer to the sink that typically carry considerably more traffic than nodes further away from the sink. The switching between the CSMA/CA and TDMA modes is controlled by the beacon messages broadcasted from the sink on demand. This makes the protocol unscalable and incurs a lot of message overhead. Moreover, the instability of nodes' MAC behavior reduces the throughput and increases the delay by a significant amount. Finally, unlike our proposed ATW-HMAC, funneling MAC handles congestions only at the region near the sink. It does not consider the congestion due to link contention at the event detection area.

Probably the most well-known and highly referenced MAC protocol for sensor network is S-MAC [5, 6]. S-MAC is a contention-based protocol that avoids overhearing and dynamically sets the duty cycle based on adaptive listening. It uses inline signaling for exchanging SYNC packets. The distributed maintenance of clock synchronization of nodes by exchanging SYNC packets and their periodic sleeps significantly reduces the throughput and increases the packet latency.

The concept of hierarchical MAC protocol was first introduced in SC-HMAC [11], where each node gets proportional medium access according to their *source count* values. It was designed for homogeneous traffic applications and single-path routing network only. It outperforms traditional MAC protocols in terms of throughput and energy efficiency. However, SC-HMAC is not suitable for diverse network applications and topologies. In the next section, we evaluate the performance of IEEE 802.11 DCF in WSN.

3 IEEE 802.11 DCF and sensor network

The operation principle of IEEE 802.11 DCF is depicted in Fig. 1, where two contending nodes i and j compete to access the medium. It is based on the CSMA/CA, where a node performs a backoff procedure before initiating the transmission of a frame. The backoff time is chosen as a random integer drawn from a uniform distribution over the interval $[0, \mathrm{CW} - 1]$, where CW is referred to as contention window. If the medium is found to be idle (i.e., it does not hear any transmission from any of its neighbors) for a DIFS period of time, the node starts to decrement its backoff time for every slot time. The node can transmit a data frame with the basic CSMA/CA procedure, or an RTS frame with the RTS/CTS procedure, when its backoff time is decremented to zero. If the medium becomes busy during a backoff process, the backoff is suspended. If the medium becomes idle again for an extra DIFS period of time, the backoff process resumes from the point where it was suspended. The contention window CW has an initial value $\mathrm{CW}_{\min}$, and is doubled when a collision occurs, up to the maximum value $\mathrm{CW}_{\max}$. For example, in 802.11b, $\mathrm{CW}_{\min}$ and $\mathrm{CW}_{\max}$ are set equal to 32 and 1,024, respectively. When a frame is successfully transmitted (i.e., ACK is received after a SIFS period of time), the contention window is set to its initial value $\mathrm{CW}_{\min}$. The value is also reset when the maximum retransmission limit is reached.

It is not efficient to apply the above MAC protocol directly in a sensor network. Since the typical traffic

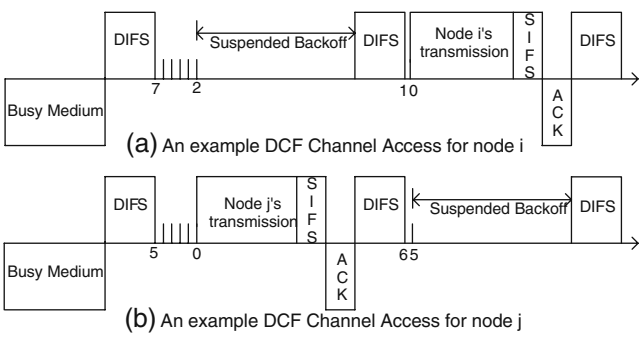
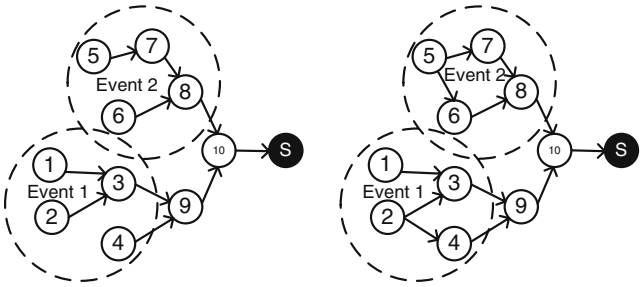


Fig. 1 IEEE 802.11 DCF operation principles



(a) Single-path routing (b) Multi-path routing

Fig. 2 An example network showing how data packets from two different events are routed from the event detection area(s) to the sink using single-path and multipath forwarding. For **b**, initially, nodes 2 and 5 divide their traffic equally towards two routes

pattern of a sensor network is many-to-one and convergent in nature [4, 11], downstream nodes have to carry more traffic than their upstreams.[1] The long-term MAC-level fairness property of the above CSMA/CA-based protocol instigates the bottleneck condition at downstream nodes. Hence, in the presence of a bottleneck node, queue build-up and packet-drop happen, which result in waste of the system resources utilized to deliver the packets halfway through the multihop network. For example, in Fig. 2a, nodes 3, 4, and 9 are in the same contention zone; only one of them can transmit at a given time, and the same is true for nodes 1, 2, 3, and 4. With CSMA/CA, the upstream sensors 1, 2, 3, and 4 collectively have more chances to send packets to node 9 than it can forward. The excessive packets received by node 9 will eventually cause *buffer overflow*, and packets are dropped. Another type of congestion happens due to media contention, namely, congestion due to *packet collision*. This is caused whenever a number of nodes contend the shared medium using the same contention parameter values, as used in the above protocol. Irrespective of the reason behind the congestion, it causes many folds of problems. When a packet is dropped/collided, the energy spent by upstream sensors on the packet is wasted. This wastage increases when the packets drop after traveling more numbers of hops. Finally, and above all, the data loss due to congestion may jeopardize the mission of the application.

We have done simulations in ns-2.30 [26] for our example network (Fig. 2a) to evaluate the performance of 802.11 DCF in a sensor network. The general simulation setup parameters are described in Section 6.2. The a/b-like values in the label of the graphs represent the data packet generations rates (i.e., traffic load) from source nodes detecting less critical and more critical events (which we define in Section 4.1 as event types A and B, respectively), respectively. The graphs of Fig. 3 illustrate that the number of packet collisions and drops greatly increases at higher traffic loads [$> 9/18$ packets per second (pps)] and, hence, the delivery ratio decreases significantly. Even the increasing number of retransmission limits cannot increase the delivery ratio to an acceptable point in case of higher traffic loads. Someone may argue that the employment of a congestion/rate control mechanism [20] might lessen the packet drops, but it can neither increase the end-to-end throughput nor decrease the packet delay significantly. In this paper, we concentrate on the role of the MAC layer in achieving the goals.

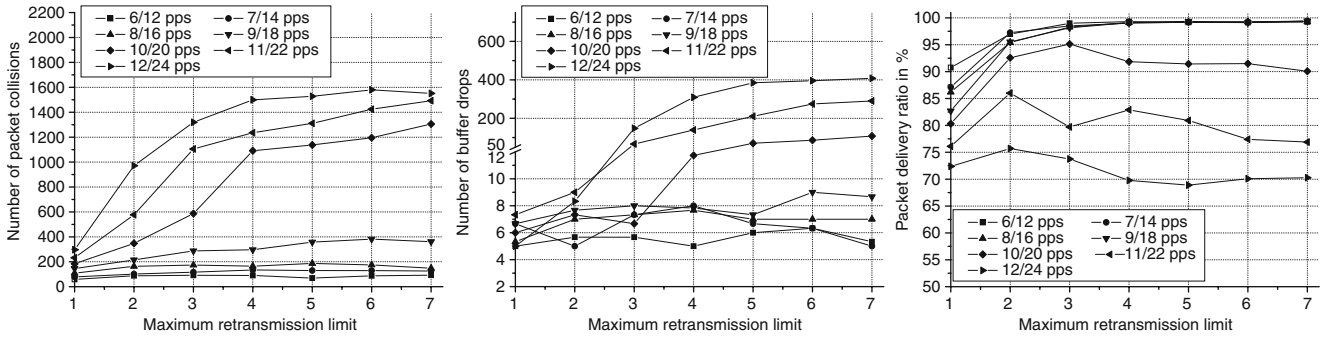
4 Proposed ATW-HMAC protocol

As described in Section 3, the use of an existing CSMA/CA-based MAC in WSN causes the source nodes to inject as many packets as possible into the network with no regard for whether the packets reach their final destination. The high number of data packets transferred by source nodes overwhelms the capacity of downstream nodes, particularly the nodes near the sink. To diminish this problem, the MAC protocol for sensor networks should be designed in such a way that the downstream nodes get more medium access than their upstreams. More explicitly, the frequency of medium access of different sensor nodes should be proportionate to the amount of data traffic carried by them. This motivation drove us to design an aggregated traffic flow weight (F^{agg}) driven hierarchical MAC protocol that gives higher access to the downstream nodes (with higher F^{agg} values) than their upstreams (with lower F^{agg} values). In what follows, we first present the network architecture and assumptions (considered for modeling and analyzing the proposed MAC protocol) in Subsection 4.1. We discuss the formulation and computation procedure of F^{agg} values for all source and forwarder nodes in Subsections 4.2 and 4.3, respectively, and then present how to determine CW_{min} values using F^{agg} in Subsection 4.4.

4.1 Network architecture and assumptions

We consider a sink-rooted, tree-based network architecture, where identical sensors are uniformly distributed

[1] We refer the parent/child relation in the sink-rooted, tree-based network as a downstream/upstream relation among the nodes. For example, in Fig. 2a, nodes 3 and 4 are the upstreams of node 9, which in turn is the downstream of them.



(a) Packet collisions experienced (b) Packet drops due to buffer overflow (c) Packet delivery ratio

Fig. 3 Performance of IEEE 802.11 DCF in WSN (**a–c**). Simulation is run for 10 s and each data point value is the mean of results from 10 runs

over the terrain. The sink may be located anywhere on the terrain. Sensor nodes and the sink are static after the deployment. A single node may be equipped with multiple types of sensors [14] (for example: humidity, temperature, accelerometer, and light sensors as in mica2, Imote2 nodes [14]), and thereby be able to detect multiple types of events. However, in the proposed traffic model, we consider that a node can only generate traffic due to one event at a given time. All nodes have equal transmission range (R_{tx}) and sensing range (R_s). We also assume that all data packets have the same size, and the buffer size of a node is counted as the maximum number of data packets that it can hold. The underlying routing protocol may use either single-path or multipath forwarding [2, 15]. In some recent research results, we observe that the multipath forwarding can achieve higher throughput, reduced end-to-end latency, higher security, and spatial energy consumption. Therefore, in the following section, we develop a framework of a traffic model for the multipath forwarding case, which can easily be tuned for single-path also. Note that, in case of multipath forwarding, the total traffic load of any node i may be distributed over a set of downstream nodes D_i, which are the next hop nodes on the routing paths from i to the sink. Similarly, if U_i represents i's set of upstream nodes, each node in U_i uses i as the next hop node on its routing path. The load distribution policy of node i towards multiple paths (defined by D_i) depends on the specific multipath routing algorithm.[2]

[2]Possible load distribution policies could be as follows: (1) homogeneous distribution—the total traffic load of node i could be equally divided amongst the available paths and (2) proportional distribution—loads could be distributed proportional to the minimum hop count of a path, free buffer space in the downstream node, maximum residual energy of the downstream node, successful packet delivery rate of a link/path, etc.

4.2 Proposed traffic model

Since all the sensors have equal sensing range, the radius of an event detection area is equal to the sensing radius of a node. Therefore, when an event occurs, a number of sensor nodes around the vicinity of the event detect it and generate data packets at a constant rate. Then, they forward the sensed data packets toward the sink in a multihop fashion. Based on the importance of the detected event, each source node i assigns a weight value w_i to the generated data packets, where $w_i \in 1, 2, 3, ..., w$. Note that $w_i = 1$ corresponds to the least important event and $w_i = w$ corresponds to the most important one. Obviously, the upper value of w determines the maximum number of traffic classifications, which is determined by the network applications. Even though the proposed traffic model and MAC protocol can work with any values of w, without loss of generality, we assume only two types of events, type A and type B.[3] Hence, the value of w_i is determined as follows:

$$w_i = \begin{cases} 1, & \textit{if node i detects an event type A} \\ 2, & \textit{if node i detects an event type B} \\ 0, & \textit{otherwise} \end{cases}$$

This weight value is determined for each event during the network initialization process and remains static for the duration of communication. In the case where all the nodes have equal weights, this value can be set to 1. Also note that the above traffic classification can easily be extended to more types, if the network applications demand.

Definition 1 (Traffic flow): A traffic flow is defined as a stream of data packets from an upstream node i to

[3]As discussed in Section 1, event type B may represent moving object tracking and A may stand for its counterpart.

a downstream node j, denoted by f_{ij}. Hence, f_{ij} may aggregate data packets from multiple flows.

Definition 2 (Aggregated traffic load): In case of multipath routing, each node divides its total traffic into multiple traffic flows, and those flows pass through multiple downstream nodes. Therefore, the aggregated traffic load of an intermediary node i (L_i) is the total sum of the data packet rates (in pps) of its all upstream traffics plus its own data packet generation rate (g_i), and is expressed as follows:

$$L_i = \Big(\sum_{k \in U_i} r_{ki} \Big) + g_i, \tag{1}$$

where r_{ki} is the data packet transmission rate (in pps) of flow f_{ki}. Note that the aggregated traffic load of a *source only node* is simply its packet generation rate, g_i.

Definition 3 (Traffic flow weight): We define the traffic flow weight of a source node i as the ratio of its instantaneous data packet reporting rate (R_i) to its data packet generation rate (g_i) times its weight value, and is expressed as follows:

$$F_i = \frac{R_i}{g_i} \times w_i \,, \; g_i > 0, \tag{2}$$

where $R_i \leq g_i$ and initially set as $R_i = g_i$ for all source nodes. The expression for R_i is carried out in Subsection 5.1. While g_i represents the application-defined constant packet generation rate for a particular event type, R_i is a dynamically controllable variable and its value is determined by an appropriate congestion control (CC) or rate control (RC) algorithm. In multipath data forwarding, a CC or RC algorithm may be activated whenever a certain path/link is congested. In this case, congestion may be reduced either by diverting the excessive amount of traffic from an overloaded path to other alternative lightly loaded paths or by decreasing the reporting rate R_i (i.e., R_i becomes less than g_i). The first option is not applicable whenever all other paths are adequately/highly loaded so that they are unable to carry additional traffic. Then, the activation of a second option becomes mandatory. However, in this paper, we take advantage of neither of the approaches in support of unbiased evaluation of the proposed ATW-HMAC protocol.[4]

[4]In our simulation, we assign $R_i = g_i$ and keep it unchanged during the whole simulation period. Obviously, the consideration of R_i opens the door of designing an optimal traffic engineering algorithm for multipath data forwarding in WSN, which we leave as our future work.

Definition 4 (Aggregated traffic flow weight): Since, in multipath routing, each downstream node may receive multiple flows from its upstreams, the aggregated traffic flow weight of any intermediate node is defined as the total sum of the weighted fractions of all flows passing through it plus its own traffic flow weight. Therefore, node i's aggregated traffic flow weight is calculated as

$$F_i^{\text{agg}} = \Big(\sum_{k \in U_i} \frac{r_{ki} \times F_k^{\text{agg}}}{L_k} \Big) + F_i. \tag{3}$$

These aggregated traffic flow weights of different sensor nodes are a new way to look into the heart of the MAC protocol for sensor network (see Section 4.4).

4.3 Distributed computation of F^{agg} values

We consider that each node k, while transmitting a data packet to its downstream node i, includes r_{ki}, L_k, and F_k^{agg} in the header of every packet. Note that a simple distributed algorithm can compute the aggregated traffic flow weights of all nodes iteratively, and it works as follows. Initially, sensor i sets $r_{ki} = 0$, $F_k^{\text{agg}} = 0$, and $L_k = 0$, $\forall k \in U_i$, and $r_{ij} = 0$, $\forall j \in D_i$. When source nodes start to transfer data packets, their downstream nodes learn the amount of incoming traffics, upstream's aggregated traffic loads, and aggregated traffic flow weights (as they are embedded into the packet headers). A downstream node i then periodically computes L_i and F_i^{agg} using Eq. 1 and Eq. 3, respectively. Note that these computations are performed only when the value of at least one parameter of the current packet differs from that of a previous one. It is obvious that L_i and F_i^{agg} values become stable whenever node i receives at least one packet from all of its upstreams. After that, in the next iteration, i's downstream nodes (j) will compute their L_j and F_j^{agg} values in the same way. This process repeats until the aggregated traffic flow weights of all nodes are calculated. Therefore, the upstream nodes would stabilize the values before their downstreams and the maximum number of iterations required before all variables become stabilized is bounded by the length of the longest routing path. More specifically, a particular intermediary node will require at most h iterations to stabilize its variables, where h denotes its hop distance from the furthest source node for which it is carrying traffic. Therefore, in our proposed ATW-HMAC protocol, the frequency of these computations is driven by the event occurrence frequency, which is typically very minimum. Also, note that the aggregated traffic flow weights of each node along the routing path(s) are updated without transferring

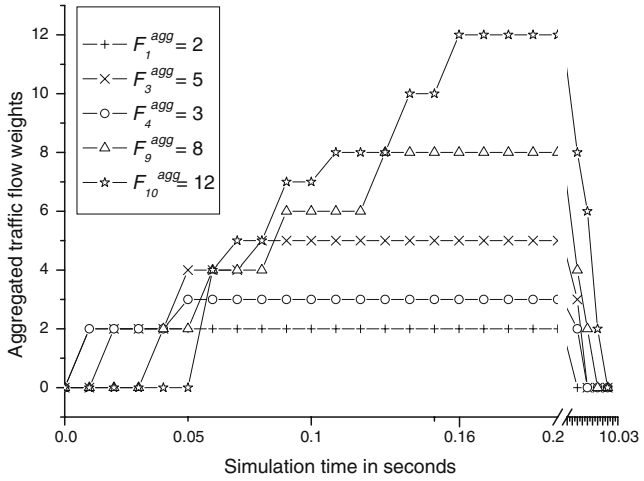


Fig. 4 F^{agg} values of nodes over the simulation time

any additional control packets, which incurs only a little overhead of a few bytes in the packet header.

The Fig. 4 plots the aggregated traffic flow weight values of five individual nodes of Fig. 2b over the simulation time, for 8/16 pps traffic load during 0.01 to 10.01 s. Values of the graphs depict that the F^{agg} values of upstream nodes get stabilized earlier than their downstreams, which is obvious. Also, the F^{agg} value of the nearest node (i.e., node 10) of the sink is stabilized at last. Note that the duration of unstable region is so small (0.16 s) that it cannot significantly affect the performance of the proposed protocol.

4.4 Determination of CW_{min} values

In order to give different levels of medium access to the nodes, we can control the contention process so that the nodes can access the wireless medium with different probabilities. The contention process of nodes can be controlled by allowing them to use different contention parameters. We can control the value of CW_{min} used by individual nodes to give them different opportunities to access the shared medium. Since the number of medium access of a node is approximately inversely proportional to its CW_{min} value [16], we can easily control each node's share of the medium access in a distributed manner using F^{agg} values. More specifically, the higher the aggregated traffic flow weight of a node, the lower its CW_{min} value is. To implement this notion, our proposed ATW-HMAC protocol gives proportional medium access to the nodes by assigning a CW_{min} value of any node i as follows:

$$CW_{min}[i] = \left\lceil \frac{(W_0 - 1) \times C}{F_i^{agg}} \right\rceil, \tag{4}$$

where W_0 is usually chosen as a power of 2, and C is a system parameter corresponding to the number source nodes within an event radius. It can be calculated as $C = \rho\pi R_s^2$ [11], where ρ is the node density and R_s is the sensing range. Therefore, the value of C linearly varies with the node density of the network. For uniform node distribution, the value of C is the same for any event irrespective of the location where the event occurs. For our deployment scenario (see Section 6.2), C's value is computed as 15.[5] The motivation of taking C value into account for the determination of CW_{min} is described below. When an event is detected, the amount of traffic produced by the sensing nodes increases with the value of C, which in turn increases the contention of the shared medium, and thus, the collisions become more frequent. This problem can be reduced by Eq. 4, which assigns higher CW_{min} values to sensing nodes taking C into consideration. In other words, the value of C helps the source-only nodes to choose more appropriate CW_{min} values that minimize collisions. For all other intermediary nodes, working as forwarders of the event traffic, the use of the same C value is justified in order to build a hierarchical medium access ordering among the nodes.

Choosing the appropriate value of W_0 is a critical concern since very higher values decrease the medium utilization and, thereby, the network throughput. Conversely, very lower value might greatly increase the medium contention, as well as packet losses due to collision. By empirical evaluations and analyzing the extensive simulation results, we found that setting $W_0 = 2^4 = 16$ produces the best results for our deployment. We also evaluated the effects of dynamic change of the W_0 value during the data traffic, but it increases the oscillations in the network and violates the fairness property. Therefore, we keep its value fixed during the whole simulation period. Also, note that W_0 is a tunable parameter depending on the node density of a network and is propagated from the sink node. Its value may be higher for low-density deployment and vice-versa, and this is required for better adjustment of CW_{min} values using Eq. 4 since C's value decreases as node density decreases. For instance, in our example network scenario (Fig. 2), $C = 4$ and we set $W_0 = 2^5 = 32$. Table 1 shows the aggregated traffic flow weights and CW_{min} values of nodes 1–10, calculated using Eq. 4.

Notice that the calculation of F^{agg} and the assignment of CW_{min} values as of Eq. 4 tells us that the nodes near the sink carry more traffic than the distant nodes, and therefore, their CW_{min} values are much smaller.

[5] $C = \rho\pi R_s^2 = \frac{1{,}000}{1{,}000 \times 1{,}000} \times \frac{22}{7} \times (70)^2 \approx 15.$

Table 1 F^{agg} and CW_{min} values of individual nodes of Fig. 2 ($R_i = g_i$, for all nodes)

Single-path Routing							Multipath Routing					
Both the events are of type A				Event 1 is B type and Event 2 is A type			Both the events are of type A			Event 1 is B type and Event 2 is A type		
Node ID	w_i	F^{agg}	CW_{min}	w_i	F^{agg}	CW_{min}	w_i	F^{agg}	CW_{min}	w_i	F^{agg}	CW_{min}
1	1	1	124	2	2	62	1	1	124	2	2	62
2	1	1	124	2	2	62	1	1	124	2	2	62
3	1	3	42	2	6	21	1	2.5	50	2	5	25
4	1	1	124	2	2	62	1	1.5	83	2	3	42
5	1	1	124	1	1	124	1	1	124	1	1	124
6	1	1	124	1	1	124	1	1.5	83	1	1.5	83
7	1	2	62	1	2	62	1	1.5	83	1	1.5	83
8	1	4	31	1	4	31	1	4	31	1	4	31
9	0	4	31	0	8	16	0	4	31	0	8	16
10	0	8	16	0	12	1	0	8	16	0	12	11

This means that the medium-access ordering is done on a per-hop basis so that any one hop node will have precedence over any two hop nodes. Therefore, the following condition holds true: $p(h_1) \geq p(h_2) \geq ... \geq p(h_s)$, $1 \leq k \leq s$, where $p(h_k)$ denotes the probability of medium access of nodes h_k hops away from the sink. The equality condition appears when two nodes have the same aggregated traffic flow weights.

The key difference of ATW-HMAC from SC-HMAC is as follows. MAC of ATW-HMAC is based on implicit traffic information. However, SC-HMAC does not calculate the amount of traffic load that an intermediary node has to forward, it merely counts the number of source nodes for which a forwarder node is carrying traffic. Therefore, it becomes unable to cope up with the dynamics of multipath data forwarding. For the same reason, even if we modify the *source count* policy of SC-HMAC a little (by counting the *source count* value 2 for event type-B nodes and 1 for type-A nodes), it lacks appropriate traffic information and thereby fails to fairly support events with varying traffic demands.

4.5 F^{agg} controlled fair packet scheduling

This subsection briefly discusses fair packet scheduling mechanisms required for fair, reliable, and guaranteed event perception. The proposed ATW-HMAC gives proportionate medium access to the nodes according to their F^{agg} values. However, it does not guarantee that an intermediate node will schedule a weighted number of packets received from different upstreams according to their F^{agg} values. The lack of this property might hinder the sink node to fairly detect an event [11], i.e., the sink node may receive quite different numbers of packets from the sensors detecting a particular event over time. More specifically, flows that traverse a larger number of hops may be penalized with respect to those having smaller hops, violating the fairness property. A weighted fair queuing [17] or weighted round robin algorithm [11] maintaining multiple queues can be practically used to guarantee fairness among all transit traffics, as well as the generated traffic. Unlike other protocols, rather than using priority index and/or buffer occupancy, in ATW-HMAC, we use F^{agg} values of incoming traffics and generated traffic as the weights for corresponding queues in each forwarder node. Notice that the above approach is also very effective in balancing end-to-end packet delays experienced by source nodes located at far distances from the sink and by those closer to the sink. The necessary mathematical developments have been carried out in the next section to prove the weighted fairness property of the proposed protocol and to measure the energy wastage.

5 Analytical analysis

In the following subsections, we present the probabilistic analysis on weighted fair rate allocation and energy consumption made by ATW-HMAC, respectively.

5.1 Weighted fair rate allocation

Property 1 (weighted fairness): An algorithm provides weighted proportional fairness *iif*, for any two nodes i and j, $F_j^{agg} = x \times F_i^{agg}$, $\forall x \in \{1, 2, 3, ..., X\} \Leftrightarrow$ the average allocated rate to j is x times higher than that to node i.

Let q_0 be the probability that a node has no packets in its queue awaiting transmission and let τ_i be the stationary transmission probability of node i in a randomly chosen time slot. A collision occurs when more than one contending nodes transmit in the same slot.

Therefore, the probability that node i's transmission results in a collision is given by

$$p_c^i = 1 - \prod_{j \in S_n, j \neq i} (1 - (1 - q_0)\tau_j), \tag{5}$$

where S_n is the set of n contending nodes. With P_{idle} denoting the probability that the medium is sensed idle during a typical slot, then, according to [18], τ_i is calculated as

$$\tau_i = b_e^{0,0} \left(\frac{(1-q_0)^2 \text{CW}_{\min}[i]}{q_0(1-p_c^i)(1-q_0^{\text{CW}_{\min}[i]})} - \frac{(1-q_0)^2 P_{\text{idle}}}{q_0} \right), \tag{6}$$

where $b_e^{0,0}$ is the probability that a node sees an empty queue after completion of postbackoff.[6] The values of p_c^i and τ_i can be computed numerically by solving the above two nonlinear equations, Eqs. 5 and 6, respectively. Since the steady state probability indicates the frequency that a node will be in a particular state in the long run, τ_i represents the fraction of data packet transmission rate of node i, i.e., $R_i \propto \tau_i$. Therefore, for any two nodes i and j, the following condition holds true:

$$\frac{\tau_i}{\tau_j} = \frac{R_i}{R_j}. \tag{7}$$

Now, let p_s^i be the probability that the transmission of a node in a randomly chosen slot is success, let DATA_i be the expected amount of data bits transmitted in the slot, and let $E[T_{\text{slot}}]$ be the average duration of slot time, θ. Then, we can express the throughput of node i as follows:

$$R_i = \frac{p_s^i \text{DATA}_i}{E[T_{\text{slot}}]}. \tag{8}$$

Considering DATA_i and $E[T_{\text{slot}}]$ are equal for all nodes in the network and neglecting collisions of more than two nodes, we can express $E[T_{\text{slot}}]$ as follows:

$$E[T_{\text{slot}}] = \sum_{i=1}^{n} \tau_i \prod_{j \in S_n, j \neq i} (1-\tau_j) T_{\text{suc}}^i + \prod_{i=1}^{n} (1-\tau_i)\theta + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \tau_i \tau_j \prod_{k \in S_n} (1-\tau_k) T_{\text{col}}^{i,j}, \tag{9}$$

where T_{suc}^i corresponds to the average duration of successful transmission from node i and $T_{\text{col}}^{i,j}$ is the duration of collisions between nodes i and j. According to [19], these two time durations are calculated as follows:

$$T_{\text{suc}}^i = T_{\text{PLCP}}^i + \frac{L_{\text{hdr}} + \text{DATA}_i}{r} + \text{SIFS} + T_{\text{PLCP}}^i + \frac{L_{\text{ACK}}}{r} + \text{DIFS}, \tag{10}$$

$$T_{\text{col}}^{i,j} = T_{\text{PLCP}}^i + \frac{L_{\text{hdr}} + \text{DATA}_i}{r} + \text{DIFS}, \tag{11}$$

where L_{hdr} and L_{ACK} are the sizes of MAC header and ACK frames in bits, respectively, and r is the channel bandwidth.[7] Now, by omitting higher-order terms of τ_i (i.e., $\tau_i^2, \tau_i^3, \ldots$) from Eq. 9, we get

$$E[T_{\text{slot}}] \approx \sum_{i=1}^{n} \tau_i T_{\text{suc}}^i + \left(1 - \sum_{i=1}^{n} \tau_i\right)\theta. \tag{12}$$

Taking derivative of Eq. 12, we have

$$\frac{\partial E[T_{\text{slot}}]}{\partial \tau_i} = T_{\text{suc}}^i - \theta \approx T_{\text{suc}}^i. \tag{13}$$

As defined in [20], proportional fair throughputs for the nodes can be achieved by the allocation that maximizes $\sum_{i=1}^{n} \log(R_i)$. The solution to this problem can be derived from

$$\frac{\partial}{\partial \tau_j}\left(\sum_{i=1}^{n} \log(R_i)\right) = 0, \forall j. \tag{14}$$

Putting the value of R_i from Eq. 8 into Eq. 14 and taking the derivative, we get

$$\left(\frac{1}{\tau_j} - \sum_{i \in S_n, i \neq j} \frac{1}{1-\tau_i}\right) E[T_{\text{slot}}] - n\frac{\partial E[T_{\text{slot}}]}{\partial \tau_j} = 0, \forall j. \tag{15}$$

For $\tau_i << 1, \forall i$, we can approximate the following:

$$E[T_{\text{slot}}] - n\tau_j \frac{\partial E[T_{\text{slot}}]}{\partial \tau_j} = 0, \forall j. \tag{16}$$

For two different nodes j and k, the above system can be rewritten as

$$n\tau_j \frac{\partial E[T_{\text{slot}}]}{\partial \tau_j} - n\tau_k \frac{\partial E[T_{\text{slot}}]}{\partial \tau_k} = 0, \forall j \neq k. \tag{17}$$

Now, combining Eqs. 13, 17, 7, and 4 yields

$$\frac{\tau_i}{\tau_j} = \frac{T_{\text{suc}}^j}{T_{\text{suc}}^i} = \frac{R_i}{R_j} \approx \frac{\text{CW}_{\min}[j]}{\text{CW}_{\min}[i]} = \frac{F_i^{\text{agg}}}{F_j^{\text{agg}}}. \tag{18}$$

[6] $b_e^{0,0}$ and P_{idle} can be expressed in terms of q_0, $\text{CW}_{\min}$, p_c^i and maximum backoff stage t; please see [18] for details.

[7] In this paper, we assume all nodes are operating in the same channel, i.e., they have equal bandwidths.

Note that the above equality implies the necessary condition for weighted fairness as defined in property 1. More specifically, the following observations can be made from the Eq. 18: *first*, the throughput ratio of two nodes i and j ($\frac{R_i}{R_j}$) is equal to the ratio of their aggregated traffic flow weights ($\frac{F_i^{\text{agg}}}{F_j^{\text{agg}}}$); *second*, the throughput of an intermediary node is the total sum of the fractional throughputs (passing through it) of its upstream nodes; and *third*, throughput of an upstream node i is $\frac{1}{x}$ times that of its downstream node j, where $x = \frac{F_j^{\text{agg}}}{F_i^{\text{agg}}}$.

5.2 Energy consumption analysis

The longevity of battery-powered sensor networks is an essential performance metric for WSN protocols. Here, we present a theoretical analysis of power consumption made by ATW-HMAC for delivering a packet from the source node to the destination sink and compare it with that of periodic sleep/schedule based MAC protocols (e.g., S-MAC, TRAMA, etc.).

5.2.1 Energy consumption analysis of ATW-HMAC

The total amount of energy consumed to deliver a single packet from source node to sink in ATW-HMAC is derived as follows. As in [21], the amount of energy consumed due to transmission of a packet from an upstream node to its downstream is calculated as follows:

$$e^{\text{trans}} = (e_{\text{te}} + e_{\text{ta}} \times R_{\text{tx}}^{\alpha}) \times L, \tag{19}$$

where e_{te} is the energy per bit needed by the transmitter electronics, e_{ta} is the energy consumption of the transmitting amplifier to send one bit over one unit distance, L is the packet size in bits, and α is the path loss factor depending on the radio frequency environment and is generally valued between 2 and 4. However, the energy per bit needed by the receiver electronics, e_{re}, does not include the energy consumption due to amplification. Therefore, the energy consumed for each packet reception is given as follows:

$$e^{\text{recv}} = e_{\text{re}} \times L. \tag{20}$$

A significant amount of energy is also consumed due to packet overhearing by the neighbor nodes. The energy consumed to read header of the packet by $(n-1)$ neighbor nodes is given as follows:

$$e^{\text{recvhdr}} = e_{\text{re}} \times L_{\text{hdr}} \times (n-1), \tag{21}$$

where L_{hdr} is the length of the header of a packet in bits. For simplicity of analysis, we do not consider the amount of energy consumed by the control packets (e.g., RTS/CTS/ACK packets) since the node energy is dissipated mainly due to the transmission and reception of data packets. However, we have to consider that the packet may not be reached at the downstream node successfully in a single transmission attempt due to collision (we do not consider link errors). We consider that a link layer ARQ mechanism is adopted at each sensor node and the maximum retry limit is set to t. If all t transmission attempts fail, then a node drops the packet.

Now, let p_{s}^i be the probability that a transmission attempt of node i is successful. The probabilistic distribution of getting first success in a repeated finite number of trials (t) is truncated geometric random [22]. Let q represent the number of transmission attempts required to transfer the packet successfully to the next hop. Using [22], we get the conditional probability mass function of q, with the condition that the transmission attempt of the node is successful, as

$$P^i(q) = \frac{(1-p_{\text{s}}^i)^{q-1} \times p_{\text{s}}^i}{1-(1-p_{\text{s}}^i)^t}. \tag{22}$$

Using geometric series equations [23], we find the expected number of transmission attempts required for single hop, $E_1[q]$, as

$$E_1[q] = \frac{1-(1-p_{\text{s}}^i)^t(1+tp_{\text{s}}^i)}{p_{\text{s}}^i(1-(1-p_{\text{s}}^i)^t)}. \tag{23}$$

Now, combining Eqs. 19, 20, 21, and 23, we get the expected total amount of energy consumed to successfully transfer a packet only one hop (for instance, the k^{th} hop) as follows:

$$\overline{E_k} = (e^{\text{trans}} \times E_k[q]) + e^{\text{recv}} + e^{\text{recvhdr}}. \tag{24}$$

If the source node of the packet is h_s hops away from the destination sink node, then the successful delivery of the packet from source to sink in ATW-HMAC consumes the following expected amount of energy:

$$\overline{E_{\text{successful}}} = \sum_{k=1}^{h_s} \overline{E_k}. \tag{25}$$

On the other hand, if the packet is dropped at any intermediate node that is h_k hops away from the source node, $h_k < h_s$, then the total amount of energy consumed by the packet up to h_k hops is wasted, and is measured as follows:

$$\overline{E_{\text{wastage}}} = \sum_{k=1}^{h_k-1} \overline{E_k} + e^{\text{trans}} \times t. \tag{26}$$

In Eq. 26, the second part of the right-hand side is included because the h_k-th hop node would drop the packet only if its t number of transmission attempts becomes unsuccessful. The value of this part must be zero if the packet is dropped due to buffer overflow.

5.2.2 Comparative discussion on energy consumption

Sensor MAC protocols designed for very low data packet rate applications can completely avoid the amount of energy consumed due to packet overhearing[8] as opposed to ATW-HMAC, shown in Eq. 21. They can further reduce the energy consumption due to *idle time listening* by switching off the radio when there is no traffic in the network. However, as the amount of traffic increases, nodes get little chance to go into sleep mode (i.e., duty-cycle increases) and the frequency of switching on–off the radio increases, which incurs energy overhead (according to [25], the energy consumed for waking up the radio once typically ranges from 8 μJ to 8 mJ).

Moreover, these protocols need to periodically broadcast special type of control packets (e.g., SYNC packets in S-MAC and scheduling packets in TRAMA) to synchronize the transmission of neighbor nodes. A greater percentage of energy is wasted by these control packets. In S-MAC, each node maintains a *schedule table* that stores the schedules of all its known neighbors and updates it periodically on hearing of SYNC packets. As data traffic increases in the network, more SYNC packets are broadcasted, which increases the collision and collapses the MAC synchronizations among the nodes. TRAMA executes neighbor protocol (NP) and schedule exchange protocol (SEP) to allow nodes to exchange two-hop neighbor information and their schedules. It also executes an adaptive election algorithm (AEA) that uses the above information to select the transmitters and receivers for the current time slot, leaving all other nodes in liberty to switch to low-power mode. Again, the energy cost incurred by NP, SEP, and AEA increases sharply as traffic increases. However, ATW-HMAC does not require any type of additional control messages, rather, it uses few bytes in packet headers for exchanging traffic flow information and thereby reduces the protocol energy overhead significantly.

The maximum amount of energy wastage happens in S-MAC and TRAMA (and in like protocols) and is due to their lack of consideration of WSN's many-to-one convergent traffic pattern. It causes huge packet drops due to collision and buffer overflow near the sink node, and therefore, the energy expenditures on those packets become useless, as measured by Eq. 26. Again, this wastage linearly increases with the increased packet generation rates of the source nodes. Therefore, for WSN applications that demand higher end-to-end throughput and lower end-to-end packet delay, the energy overhead of periodic sleep–listen MAC protocols is much higher than that of ATW-HMAC. The next section shows the supportive results found in our simulation.

[8] In some cases, overhearing is indeed desirable. Some algorithms may rely on overhearing to gather neighborhood information for network monitoring, reliable routing, or distributed queries [11, 24].

6 Performance evaluation

The effectiveness of the proposed ATW-HMAC protocol is evaluated through packet-level simulation of traffic flows in ns-2.30 [26], with varying packet generation rates.

6.1 Performance metrics

We compare IEEE 802.11DCF, S-MAC [6], TRAMA [3], and SC-HMAC [11] with ATW-HMAC, in terms of the following performance metrics:

- *Throughput*: We measure the end-to-end effective data throughput in kilobytes per second for different traffic loads. The results do not count any control packets. Only data packets received at the sink are counted for the throughput calculation. The higher the value is, the better the protocol performance is.
- *End-to-end packet delay*: Delay of a single packet is measured as the time difference between when the packet is received at sink and its generation time at the source node. Delays experienced by individual data packets are averaged over the total number of individual packets received by the sink. The lower the value is, the better the performance is.
- *Delivery ratio*: Delivery ratio is the ratio of the total number of packets received by the sink to the number of packets generated by all the source nodes. We express it in percentage. The higher the value is, the better the protocol performance is.
- *Energy cost per packet*: We have measured the total amount of energy dissipated by all source and forwarder nodes during the simulation time and took the average. We then divide this per-node average energy consumption by the number of packets received by the sink. Therefore, if the same amount

of total energy is dissipated by any two protocols, then the protocol that has a higher delivery ratio will perform better than the other one.

- *Total energy wastage*: Whenever a packet is dropped before reaching the destination sink, the amount of energy expenditures on the packet by the source and forwarder nodes is considered as wastage (see Eq. 26). We have added up all these wastages during the whole simulation period to quantify the performance of the protocols.
- *Protocol operation overhead*: Protocol operation overhead is the ratio of the total number of control bytes[9] transmitted to the total number of data packet payload bytes and control bytes transmitted by all nodes during the whole simulation period. We express it in percentage.
- *Protocol energy overhead*: We have measured the total amount of energy dissipated due to the transmission and reception of control bytes by all source and intermediary nodes during the whole simulation period to evaluate and compare the energy overhead of different protocols.
- *Weighted fair rate allocation*: We measure the end-to-end effective data throughput for two events with varying packet generation rates to evaluate fair rate allocation. We compare the fairness of the proposed protocol only with 802.11 MAC and SC-HMAC.

6.2 Simulation setup

The simulation environment parameters are listed in Table 2. Other than the default values in 802.11 MAC layer implementation in ns-2.30, we have done the following changes: the MAC header size is increased to 184 bits (additional 24 bits for inserting r_{ij}, L_i, and F_i^{agg} fields in each packet), and the value of CW_{min} of each node is determined online according to ATW-HMAC protocol. We establish p number of alternate braided paths (where $1 \leq p \leq 3$) from each node to the sink using [27]. We consider that paths do not change/fail during data traffic and each upstream node distributes its aggregated traffic load equally towards all of its downstream nodes. In the implementation of S-MAC, we use a dynamic duty cycle (1% to 99%); 115-ms-duration listen interval; SYNC packet size of 10 bytes; and contention window values of 15 and 31 time slots for SYNC packets and data packets, respectively. Similarly, for implementing TRAMA, we fix up the maximum size of the signaling packet at 128 bytes and schedule exchange packets at 21 bytes, as used in [3]. However, unlike in [3], we allow each node to compute *SCHEDULE_INTERVAL* based on the rate at which packets are produced by the higher-layer application, ranging from 50 slots to 800 slots with 50-slot intervals. For the sake of fair comparison of ATW-HMAC with SC-HMAC, we also modify *source count* policy of SC-HMAC as stated in the last paragraph of Section 4.4.

Unless otherwise specified, we execute the simulation runs for four different event traffics: two A-type events (A1 and A2) and two B-type events (B1 and B2). The event location, duration, and the data packet generation rates of event source nodes are listed in Table 3. Note that the events join and leave the system at random time instants, and some of them are time-overlapped. We execute 10 simulation runs and take the average in order to show more stable values for each data point in the result graphs.

Table 2 Simulation parameters

Parameter	Value
Terrain area	1,000 × 1,000 m
Number of sensor nodes	1,000
Node distribution	Uniform random
Trans. radius, R_{tx}	100 m
Sensing radius, R_{s}	70 m
Initial node energy	50 J
Channel bandwidth, r	512 kbps
Buffer size	30 packets
Payload size	64 bytes
PHY, MAC headers	192 and 184 bits, respectively.
RTS, CTS, ACK pkts.	160, 112, and 112 bits, respectively.
T_{slot}, DIFS, SIFS	10, 34, and 10 μs, respectively.
Max. retry limit, t	4
Transmit power	$7.214e^{-3}$ W
Rcv. signal threshold	$3.65209e^{-10}$ W
Radio wake up energy	1 mJ
Sleep power	15 μW
Propagation model	TwoRayGround
PHY error model	Uniform random
Bit error rate	10^{-3}
FEC strength	1
Simulation time	30 s
Sink	1 at location [1,000, 500]

Table 3 Event and traffic settings

Event	Location	Duration	Packet gen. rate
A1	(200, 500)	7^{th}–18^{th} s	1–8 pps
A2	(400, 200)	12^{th}–25^{th} s	1–8 pps
B1	(700, 150)	5^{th}–10^{th} s	2–16 pps
B2	(600, 800)	19^{th}–22^{nd} s	2–16 pps

[9] SYNC, RTS, CTS, and ACK packets in S-MAC; NP and SEP messages in TRAMA; and RTS, CTS, ACK, and additional MAC header bytes in SC-HMAC and ATW-HMAC are considered for counting the total number of control bytes transmitted by the respective protocol.

6.3 Simulation results

Figure 5 shows the performance comparisons of five different MAC protocols in terms of throughput, delay, delivery ratio, energy cost, energy wastage, and protocol operation overhead. The a/b-like values in the X axis of the figures represent the packet generation rates of event types A and B, respectively. We observe from the graphs of Fig. 5a that, for lower data packet generation rates ($\leq 3/6$ pps), the effective data throughput of the proposed protocol is not significantly increased from that of 802.11 MAC. This is because, at lower rates, the aggregated traffic load of nodes near the sink is not high enough to make them congested. However, the gap increases with the increased traffic loads. As in S-MAC, TRAMA, and IEEE 802.11 protocols, the average end-to-end packet delay greatly increases with higher traffic loads, hence, the throughput decreases. Alternatively, the hierarchical MAC protocols can operate at higher-throughput regions since they can reduce both the packet drops due to funneling effect [4] and end-to-end packet delay by a considerable amount. At higher rates, ATW-HMAC defeats SC-HMAC in terms of achievable throughput because nodes in SC-HMAC lack the appropriate traffic information (especially for multipath forwarding case) and, hence, fail to choose optimal contention parameters. For ATW-HMAC, the network reaches saturation condition at about 7/14 pps traffic load, i.e, further increase of load cannot raise the throughput, rather, it decreases.

According to graphs of Fig. 5b, as source nodes increase their packet generation rates, the end-to-end packet delay increases in all protocols. However, packets in ATW-HMAC reach their final destination in minimum delay (less than 280 ms for 8/16 pps) and more than two times faster than those in 802.11 MAC. The main reason behind this result is that the latter protocol gives almost equal medium access to all neighbor nodes (irrespective of their traffic loads), which causes the packets to experience a noteworthy amount of delay at each intermediate node. It is also observed that SC-HMAC packets are much slower than ATW-HMAC packets for higher packet rates and the delays experienced by other protocols are really incomparable. The other reason for obtaining the above results is that, when traffic converges near the sink, collision increases greatly in other protocols that force a node to retransmit a single packet many times, which in turn increases the queuing delays of packets.

The graphs of Fig. 5c tell us that, as the packet generation rate increases at the source nodes, the delivery ratio falls suddenly in case of nonhierarchical MAC protocols. This happens due to the funneling effect [4] near the sink node, which causes a lot of packet

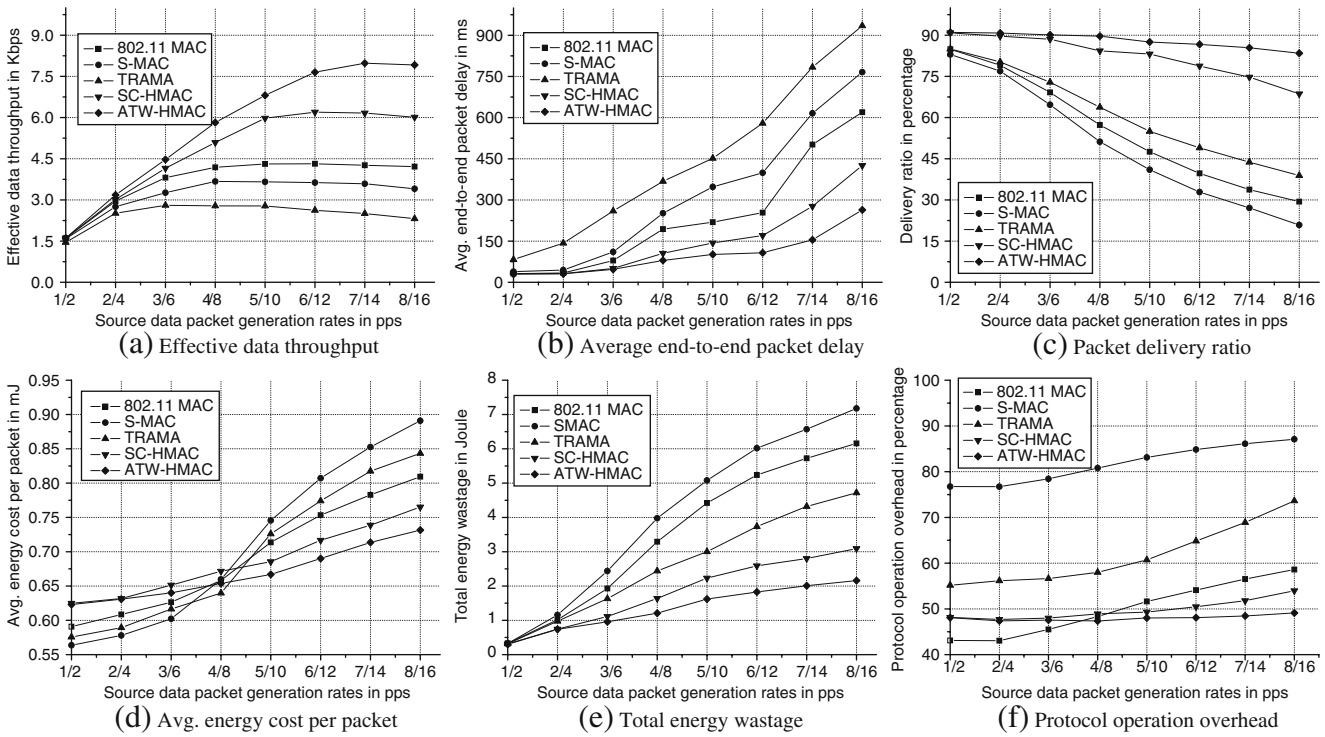


Fig. 5 Performance comparisons of five different MAC protocols (**a**–**f**)

drops due to collision and buffer overflow. However, ATW-HMAC can guarantee more than 85% packet delivery even at very high rates (8/16 pps). SC-HMAC provides a comparatively worse delivery ratio due to its incapability of taking the full advantage of multipath routing.

Figure 5d shows the average energy cost per successful packet reception. The graphs show that, for lower packet generation rates (up to 4/8 pps), the average energy cost of S-MAC and TRAMA is much lower than the hierarchical MAC protocols, and this is due to their periodic sleep/schedule mechanisms. However, as their duty cycle increases and the delivery ratio decreases with higher data packet rates, the energy cost rises to higher values. This result claims that the periodic sleep/schedule mechanisms provide better performance if the input traffic rate of the network is kept much lower than the channel capacity. On the other hand, for higher data packet rates, ATW-HMAC outperforms all other protocols by a significant amount in terms of average energy cost per packet, and this is obtained due to its capability of achieving higher delivery ratio. Also, the energy overhead for protocol operation is much less in ATW-HMAC as compared to S-MAC and TRAMA.

The total energy wastages due to the dropped packets during the whole simulation period are plotted in Fig. 5e. As expected, the energy wastage sharply rises up at higher traffic loads in nonhierarchical MAC protocols, whereas in SC-HMAC and ATW-HMAC, it increases very slowly. The reason why TRAMA protocol gives better performance than 802.11 in terms of energy wastage is stated as follows: TRAMA has collision-free data transmission slots. Even though it has significant energy overhead due to protocol operation (i.e, for broadcasting NP and SEP messages, see [3]), it greatly reduces the additional energy consumption due to RTS/CTS or data packet collisions. ATW-HMAC offers minimum energy wastage for all traffic loads; the cause behind this is already explained in the previous paragraph.

In graphs of Fig. 5f, we observe that the protocol operation overheads of S-MAC and TRAMA protocols are much higher than those of others. This is due to additional overheads introduced by SYNC and NP/SEP messages, respectively. At lower packet generation rates, 802.11 MAC gives better performance than hierarchical MAC protocols since it does not use control bytes in the packet headers. However, as the number of collisions in 802.11 MAC significantly increases at higher rates, its performance degrades by a considerable amount. Finally, ATW-HMAC provides almost the same performance of SC-HMAC at lower packet generation rates and slightly better performance at higher rates. This is caused by more packet collisions in SC-HMAC. In Fig. 6a, we plot the total amount of energy expenditures due to transmission and reception of control bytes in different protocols during the whole simulation period. We see that, as traffic generation rate increases, the energy consumption overhead of all protocols increases sharply. Nevertheless, as expected theoretically, at higher traffic loads, the energy overhead of S-MAC and TRAMA protocols are considerably higher than the hierarchical protocols, even though they perform slightly better at low traffic.

Figure 6b compares performance of the proposed ATW-HMAC protocol with that of SC-HMAC and 802.11 MAC only in terms of event throughput. For the data point values of graphs in Fig. 6b, we execute simulation runs for two events only: A2 and B2, during

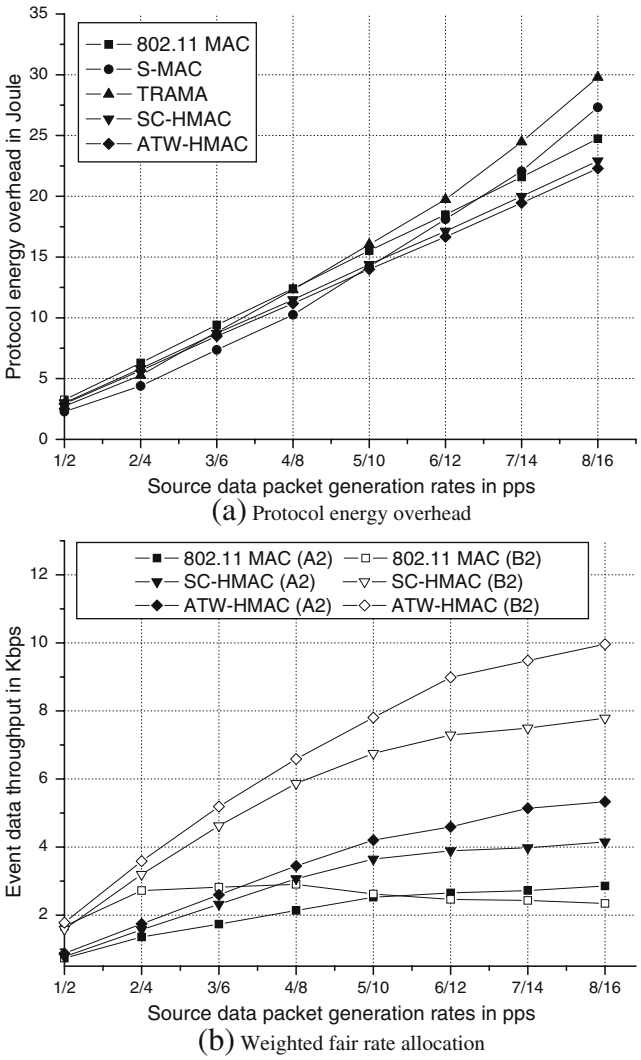
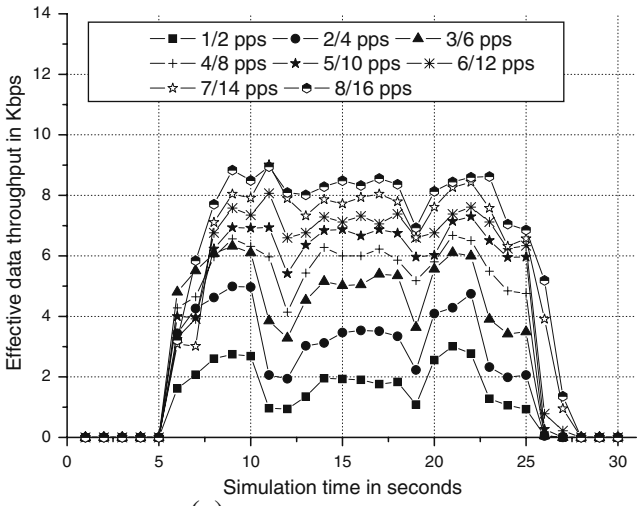


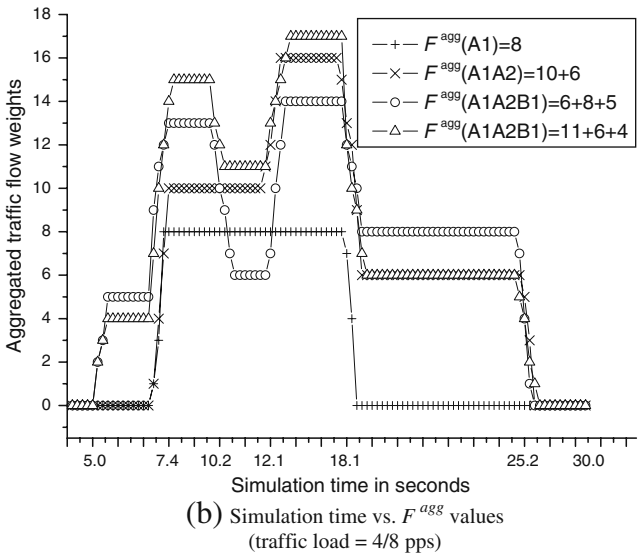
Fig. 6 Energy overhead and event fairness (**a**, **b**)

the 5th–10th seconds. As expected theoretically, we reveal from the graphs that throughput of event type B in 802.11 MAC is almost double that of type A at lower traffic rates. However, as the traffic rate increases, B's throughput decreases, and it even becomes less than A's throughput, due to increased collisions and packet drops. On the other hand, B's throughput in ATW-HMAC is almost double that of A's throughput for all traffic rates, and it outperforms SC-HMAC also.

In Fig. 7a, we plot the end-to-end effective data throughput of ATW-HMAC protocol achieved during the whole simulation period for varying packet generation rates, at 1-s intervals. We see that, in each of the graphs, the highest throughput is achieved during the 8th–11th seconds and 20th–23rd seconds, since the network simultaneously carries both A- and B-type event traffics during these periods. However, in the graphs for lower traffic loads ($\leq 3/6$ pps), we observe fewer throughputs at the 11th and 23rd seconds. Many of the generated packets reach the destination sink within the second in which they were generated. We also observe that the throughput for the traffic load 2/4 pps is almost double that for 1/2 pps. Nevertheless, the throughput improvement ratio decreases as the input packet rate increases. This result is also congruent with that found in Fig. 5a.

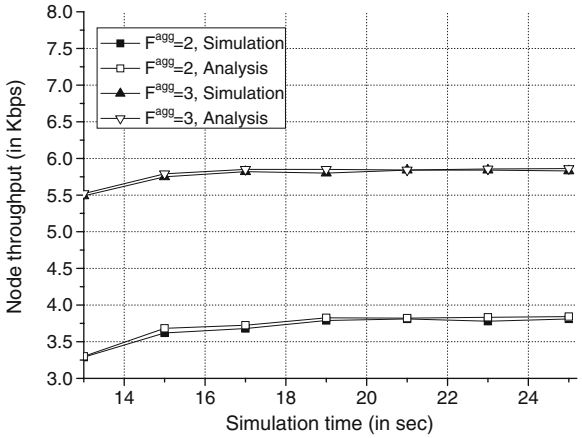


(a) Simulation time vs. throughput

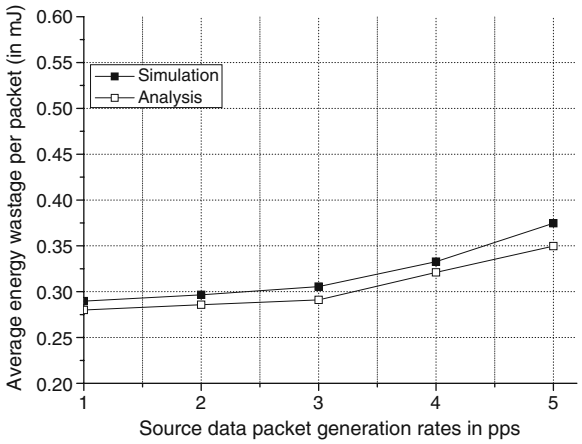


(b) Simulation time vs. F^{agg} values (traffic load = 4/8 pps)

Fig. 7 Throughput and F^{agg} values over the simulation time (**a**, **b**)



(a) Simulation time vs. node throughput (traffic load = 4pps)



(b) Energy wastage per packet for varying traffic loads

Fig. 8 Performance comparisons between analysis and simulation results (**a**, **b**)

Like in Fig. 4, we plot the aggregated traffic flow weights of four different nodes over the simulation time in Fig. 7b. $F^{agg}(A1A2) = x + y$-like values in the label of the figure represent that the chosen node carries x unit traffic weight of event A1 and y unit of event A2. We observe that the traffic weight update period of the nodes at the joining/leaving time of an event is much

less (as maximum as 0.46 s) than the event duration. We also see that the nodes carrying higher amounts of traffic weights take a little more time to stabilize their variables than others carrying small amounts of traffic.

Finally, we have executed a set of different simulation runs for the event A2 traffic only in order to show comparisons between performances of the analysis and the simulation results. We have measured the throughputs achieved by one upstream and one downstream node of a routing path having aggregated traffic flow weights 2 and 3, respectively. For producing the analytical results, we have used parameter values from Table 2 and set $q_0 = 0.07$ for the upstream node and $q_0 = 0.04$ for the downstream node. The results from 10 runs are averaged and plotted in Fig. 8. The graphs of Fig. 8a show that nodes can maintain weighted fairness both in simulations and analysis. Analysis gives slightly better results since we did not consider the bit errors there. In Fig. 8b, we plot the average energy wastage per packet for varying packet generation rates. We observe that the wastage slowly increases with the packet generation rates. This is because of increasing collisions (or decreasing success probability, we varied p_s^i in between 0.95 and 0.65) at higher traffic loads, which in turn increase the retransmissions and packet drops. The observed energy wastage in analytical results is a bit less than the simulation results because we did not consider computation costs in the analysis.

7 Conclusions

To the best of our knowledge, the proposed ATW-HMAC protocol is the first approach that exploits both the event traffic priorities and convergent traffic pattern in designing a MAC protocol for sensor networks. It offers higher end-to-end effective data throughput, reduced packet delay, and increased packet delivery ratio by assigning the minimum contention window value (CW_{min}) in hierarchical order based on the weight of traffic a node carries. The results also show that ATW-HMAC finds its utility in applications where expected data packet generation rates are medium to high.

How to further increase the overall throughput, whenever data packets from a single source node are routed over multiple paths, by employing a dynamic traffic engineering algorithm, which would work on top of ATW-HMAC, is the next problem to address. We also plan to extend the proposed protocol so that it can work with satisfactory performance in random (or unknown) node distribution environment.

Acknowledgements The authors thank the anonymous reviewers for their thoughtful comments and suggestions that helped to clarify this work's presentation. This research was supported by the MKE, Korea, under the ITRC support program supervised by the IITA (IITA-2009-(C1090-0902-0002)). Dr. C. S. Hong is the corresponding author.

References

1. Akyildiz IF, Molodia T, Chowdhury KR (2007) A Survey on wireless multimedia sensor networks. Comput Networks 51(4):921–960
2. Razzaque MA, Alam MM, Rashid MM, Hong CS (2008) Multi-constrained QoS geographic routing for heterogeneous traffic in sensor networks. IEICE Trans Commun E91B(8)
3. Venkatesh R, Katia O, Garcia-Luna-Aceves JJ (2006) Energy-efficient, collision-free medium access control for wireless sensor networks. Wirel Netw 12(1):63–78
4. Gahng-Seop A, Emiliano M, Campbell AT, Hong SG, Francesca C (2006) Funneling-MAC: a localized, sink-oriented MAC for boosting fidelity in sensor networks, In: Proc. of 4th ACM conference on embedded networked sensor systems (SenSys 2006), Boulder, 1–3 November 2006
5. Ye W, Heidemann J, Estrin D (2002) An energy-efficient MAC protocol for wireless sensor networks. In: Proc. of IEEE INFOCOM. IEEE, New York, pp 1567–1576
6. Ye W, Heidemann J, Estrin D (2004) Medium access control with coordinated adaptive sleeping for wireless sensor networks. IEEE/ACM Trans Netw 12(3):493–506
7. van Dam T, Langendoen K (2003) An adaptive energy-efficient MAC protocol for wireless sensor networks. In: Proc. of 1st ACM conference on embedded networked sensor systems (SenSys). ACM, Los Angeles, pp 171–180
8. Changsu S, Young-Bae K (2005) A traffic aware, energy efficient MAC protocol for wireless sensor networks. IEEE Int Symp Circuits Syst (ISCAS 2005) 2975–2978
9. Venkatesh R, Garcia-Luna-Aceves JJ, Katia O (2005) Energy-efficient, application-aware medium access for sensor networks. In: Proc. of IEEE international conference on mobile adhoc and sensor systems (MASS), Washington, DC, 7–10 November 2005
10. Sahoo A, Baronia P (2007) An energy efficient MAC in wireless sensor networks to provide delay guarantee. In: Proc. of 15th IEEE workshop on local and metropolitan area networks, 10–13 June. IEEE, Piscataway, pp 25–30
11. Rashid MM, Alam MM, Razzaque MA, Hong CS (2007) Congestion avoidance and fair event detection in wireless sensor networks. IEICE Trans Commun E90-B(12):3362–3372
12. Wu H, Cheng S, Peng Y, Long K, Ma J (2002) IEEE 802.11 distributed coordination function (DCF): analysis and enhancement. In: Proc. of IEEE ICC 2002, vol 1, New York, 28 April–2 May, pp 605–609
13. Chou CT, Shin KG, Shankar S (2006) Contention-based airtime usage control in multirate IEEE 802.11 wireless LANs. IEEE/ACM Trans Netw 14(60)
14. Crossbow (2008) Mica2/Imote2 Mote Datasheet. http://www.xbow.com
15. Baek SJ, Veciana GD (2007) Spatial energy balancing through proactive multi-path routing in wireless multi-hop networks. IEEE/ACM Trans Netw (TON) 15(1):93–104

16. Bianchi G (2000) Performance analysis of the IEEE 802.11 distributed coordination function. IEEE J Select Areas Commun 18(3):535–547
17. Branchs A, Perez X (2002) Distributed weighted fair queuing in 802.11 wireless LAN. In: IEEE int'l conf commun (IEEE ICC). IEEE, New York, pp 3121–3127
18. Malone D, Duffy K, Leith D (2007) Modeling the 802.11 distributed coordination function in nonsaturated heterogeneous conditions. IEEE Trans Netw 15(1):159–172
19. Banchs A, Serrano P, Oliver H (2007) Proportional fair throughput allocation in multi-rate IEEE 802.11e wireless lANs. Wirel Netw J 13(5):649–662
20. Kelly FP, Maulloo AK, Tan DKH (1998) Rate control in communication networks: shadow prices, proportional fairness and stability. J Oper Res Soc 49(3):237–252
21. Tang S, Li W (2006) QoS supporting and optimal energy allocation for a cluster based wireless sensor network. Comput Commun 29(13):2569–2577
22. Bertsekas D, Gallager R (1992) Data networks, ex. 2.13, 2nd edn. Prentice Hall, Englewood Cliffs
23. Yates RD, Goodman DJ (2005) Probability and stochastic processes, B.6. Wiley, New York
24. Madden S, Franklin MJ, Hellerstein JM, Hong W (2002) TAG: a tiny agregation service for Ad-Hoc sensor networks. Presented at the 5th Symp. Operating systems design and implementation (OSDI), Boston, 9–11 December 2002
25. Yibei L, Chung-Min C (2007) Energy saving via power-aware buffering in wireless sensor networks. IEEE INFOCOM 2411–2415, Anchorage, 6–12 May 2007
26. Network Simulator (2009) The Network Simulator - ns-2.30. http://www.isi.edu/nsnam/ns/ns-build.html
27. Ganesan D, Govindan R, Shenker S, Estrin D (2001) Highly-resilient, energy-efficient multipath routing in wireless sensor networks. In: Proc. of the 2nd ACM international symposium on mobile ad hoc networking and computing, Long Beach, 4–5 October 2001