TLS: traffic load based scheduling protocol for wireless sensor networks

Prasan Kumar Sahoo*

Department of Computer Science and Information Engineering,
Chang Gung University,
Taoyuan, 33302, Taiwan

and

Division of Colon and Rectal Surgery,
Chang Gung Memorial Hospital,
Linkou, Taiwan
Email: pksahoo@mail.cgu.edu.tw
*Corresponding author

Hiren Kumar Thakkar

Department of Electrical Engineering,
Division of Computer Science and Information Engineering,
Chang Gung University,
Taoyuan, 33302, Taiwan
Email: D0221015@stmail.cgu.edu.tw

Abstract: In wireless sensor networks (WSNs), nodes are usually deployed in the monitoring region randomly and densely and are supposed to monitor the region for a longer duration. These sensors are normally powered by a battery and therefore it is essential to regulate the power utilisation of the nodes efficiently. Although most of the current protocols reduce the power utilisation by regulating the sleep and wake-up schedules, they fail to make an adaptive sleep or wake up schedule for the nodes based on their traffic load. This paper proposes a traffic load based adaptive node scheduling protocol to determine the active and sleep schedules of the nodes. The entire network is partitioned into a set of virtual zones and a routing path selection algorithm is proposed considering the residual power of the next hop nodes. Simulation results show that the energy consumption and packet overhead of our protocol are considerably less as compared to similar quorum-based medium access control (MAC) protocols.

Keywords: WSNs; wireless sensor networks; MAC protocol; scheduling.

Reference to this paper should be made as follows: Sahoo, P.K. and Thakkar, H.K. (2019) 'TLS: traffic load based scheduling protocol for wireless sensor networks', *Int. J. Ad Hoc and Ubiquitous Computing*, Vol. 30, No. 3, pp.150–160.

Biographical notes: Prasan Kumar Sahoo received MSc in Mathematics from Utkal University, India in 1994, MTech in Computer Science from IIT, Kharagpur, India in 2000, first PhD in Mathematics from Utkal University, and second PhD in Computer Science and Information Engineering from National Central University, Taiwan, in 2002 and 2009, respectively. He is a Professor in the Department of Computer Science and Information Engineering, Chang Gung University, Taiwan. He is also an Adjunct Researcher in the Division of Colon and Rectal Surgery, Chang Gung Memorial Hospital, Linkou, Taiwan. His current research interests include big data analytic, and cyber-physical systems. He is IEEE Senior Member.

Hiren Kumar Thakkar received his BE in Computer from A.D. Patel Institute of Technology, India in 2009, and the MTech in Computer Science and Engineering from IIIT Bhubaneswar in 2012. He has completed PhD from the Department of Electrical Engineering, Division of Computer Science and Information Engineering, Chang Gung University, Taiwan in 2018. He is currently working as a Postdoctoral research fellow at Motor Behavioral Research Lab at Healthy Aging Research Center of Chang Gung University, Taiwan. His research interests include the areas of bio-medical big data analysis, prediction, optimization, and machine learning.

1 Introduction

In recent years, micro-electro-mechanical systems (MEMS) has revolutionised the industrial as well as consumer products with embedded sensing technology. These small sized devices are used in wireless sensor networks (WSNs) (Rault et al., 2014) due to their inexpensiveness, multi-functionality and self-configuration characteristics. MEMS is one of the most promising technologies, which is used for diversified applications such as health-care monitoring, environmental monitoring, military surveillance, mobile object navigation and tracking. Normally, these tiny wireless sensor nodes are deployed densely and randomly over any monitoring region and are self-organised to form the network. After deployment of the sensors, each of them is supposed to aggregate, store, compute and transmit data to the sink in an event driven or time driven manner to complete its own mission.

Limited to power, processing and memory constraints, these tiny nodes always coordinate with its one-hop neighbours to transmit the data to the sink in a multi-hop fashion. In general, these nodes are battery powered and are deployed over harsh terrains, where it is difficult even if impossible to recharge and replenish the battery. In the WSN, sensors may die predictably due to exhaustion of power and unpredictably due to technical and software failure. In the post-deployment scenario, if a large number of sensors are drained out of power, the network must recycle and redeploy them. The recycling and redeployment of sensors not only increase the monetary cost considerably but also affect the purpose of the deployment as the dead sensors can no longer collect the data anymore. Hence, it is highly essential to develop efficient algorithms to minimise the consumption of power and let the network lifetime be extended.

Scheduling of the node and selection of efficient medium access control (MAC) protocols play important roles to improve the energy efficiency of the sensors and thereby the network lifetime. Though, nodes should go to the power saving modes to improve the network lifetime, a subset of sensors should be scheduled efficiently so that some nodes can remain active for a specified amount of time to ascertain the connectivity and the coverage. However, due to a characteristic of deployment of sensors, collision is a challenging issue in WSNs, which occurs when two or more nodes transmit data packets at the same time frames over the shared transmission medium. Hence, it is important to control the access of shared medium in WSNs, which can facilitate the smooth functioning of the network. Usually, MAC protocols are designed to help each node to determine how and when to access the shared channel. This issue is also known as channel allocation or multiple access problems.

The MAC layer is a part of the data link layer and the primary goal of designing an efficient MAC protocol is to reduce the number of collisions from the neighbouring nodes. Besides, to design a MAC protocol with low duty cycles for a dense deployment of sensors is also a challenging problem, which needs to be addressed in an energy-efficient manner. Moreover, it is observed that communication in sensor network consumes more energy as compared to the computation and therefore it is important to reduce the energy cost of communication to accomplish the desired network operations. Normally, a power-efficient MAC protocol should conserve a significant amount of battery power with proper sleep and wake up schedules in place. Hence, in this paper our primary focus is to design a power-efficient MAC protocol to prolong the network lifetime by balancing the load on each node.

Many studies have identified the idle listening as a major reason behind the wastage of node power (Peng et al., 2017; Heinzelman et al., 2016; Ye et al., 2004). A simple and intuitive solution to extend the lifetime of the network is to engage nodes in a duty-cycled manner with alternate sleep/wake up schedule at regular time intervals. While synchronisation of such sleep/wake up schedules is a challenging problem, irrationally longer sleep periods may increase the response time and lower the effectiveness of the sensors. In Zheng et al. (2005), authors have proposed mechanisms by which sensors can determine the predefined active/sleep mode based on the traffic load to adjust the idle listening period of the nodes. In this case, at the beginning of each time frame, the protocol has to check whether it needs to receive or transmit any pending data or not. However, the fixed sleep/wake up design may waste the power unnecessarily if the network has light traffic and also causes an additional delay for heavy traffic. Hence, it is necessary to reduce the duration of the time that a sensor node spends in idle listening.

The rest of the paper is organised as follows. Related works are given in Section 2 and problem formulation along with system model are discussed in Section 3. The node scheduling protocol based on the dynamic traffic load of a node is described in Section 4. Theoretical analysis of the impact of different parameters on the proposed protocol in terms of latency is studied in Section 4.4. Performance evaluation of our proposed protocol is carried out in Section 5 followed by conclusion in Section 6.

2 Related work

In past, there have been many efforts to extend the lifetime of the network such as power-efficient MAC protocols (Malekshan et al., 2014; Wu et al., 2014; Chou et al., 2013; Sherly Puspha and Murugan, 2015; Hwang et al., 2013; Chou et al., 2011; Ye et al., 2004; Van Dam and Langendoen, 2003; Zheng et al., 2005; Chao and Lee, 2010; Xi et al., 2010; Sahoo et al., 2012), routing protocols (Anisi et al., 2013; Zytoune et al., 2010; Sohrabi et al., 2000) deployment protocols (Heinzelman et al., 1999; Olariu and Stojmenovic, 2005) and node scheduling protocols (Chao et al., 2013; Ha et al., 2006).

The sensor-MAC popularly known as SMAC (Ye et al., 2004) tries to reduce the idle listening of sensor nodes by asking them to sleep at regular intervals, if nodes are not engaged in any sorts of communication activities. The idea of SMAC is identical to that of IEEE 802.11 power conserving mode, where nodes wake up in the beginning of each beacon interval and check whether to remain active or sleep again. Since SMAC keeps the duty cycle low, it is able to reduce the significant power consumption of the nodes. However, SMAC protocol has few demerits. First of all, SMAC may incur longer transmission time, i.e., latency due to its low duty cycle. Moreover, due to the fixed duty cycle, it fails to adapt to the dynamic network traffic load. For example, if the duty cycle is decided based on the node with heavy traffic, an ample amount of power may be wasted for lightly loaded nodes. On the contrary, if the duty cycle is decided based on the lightly loaded nodes, the higher transmission time is expected.

The timeout-MAC also known as TMAC (Van Dam and Langendoen, 2003) has an improvement over SMAC and it uses an adaptive duty cycle. In TMAC, sensor nodes go to sleep mode until and unless they have pending data packets to send and receive up to certain time-out interval T_A. Although TMAC proposes a novel method to determine the active duration of the nodes, it still has the disadvantage of long transmission time. Interestingly, similar to IEEE 802.11 power conserving mode, both SMAC and TMAC protocols mandate all sensor nodes to wake up at the beginning of each time frame leading to the wastage of power, since light-loaded nodes may continue to remain idle in most cycles.

In the pattern-MAC i.e., PMAC (Zheng et al., 2005), authors propose a method by which nodes can dynamically regulate the listening period and can decide the sleep/wake up mode considering their dynamic traffic condition. During heavy traffic load, nodes may continue to remain in the active mode for a longer period of time and keep on checking whether to transmit/receive data at the beginning of each time frame. Contrary to this, nodes may choose to prolong their sleep mode and may not require to wake up to transmit/receive data at each time frame at the times of light traffic load. Once sensors determine their sleep/wake up schedules, the scheduling information is exchanged among neighbour nodes and the same is coordinated by the PMAC. One condition is needed in this method is that it must guarantee the information can be received from each sensor.

Quorum-based MAC (QMAC) protocols (Chen et al., 2015; Wu et al., 2011; Chao and Lee, 2010; Xi et al., 2010) vary from the SMAC, TMAC and PMAC to avoid compulsory wake up of nodes at every instant of time and employ the quorum concept. In the recent past, the quorum theory has many diversified applications in the field of distributed systems, which provides mutual exclusion guarantees, agreement, voting and fault tolerance. This paper employs the quorum set to know the time frames during which sensors have to wake up. There are various kinds of quorum such as grid-based (Cheung et al., 1992), majority-based (Thomas, 1979), tree-based (Agrawal and El Abbadi, 1991) etc. Without loss of generality, we use the grid-based quorum to design the proposed protocol. In a grid-based quorum, one row and one column are chosen randomly from a $G \times G$ grid, which according to the property of quorum ensures the meeting of any two sensors in at least one-time frame (Chao and Lee, 2010).

In this paper, a traffic load based adaptive node scheduling protocol is proposed to counter the above-mentioned disadvantages. The main idea is to develop a node scheduling protocol that uses a traffic load model combined with grid-based quorum mechanism. In the traffic load model, each sensor will estimate its own accumulated traffic load based on the density of the nodes located within its communication range and the hop counts from the sink node. Thus, it can set the traffic load dynamically based on the number of active and dead nodes to decide its active and inactive scheduling.

3 Problem formulation

Wireless sensor networks are primarily used for monitoring remote regions such as forest, country borders, mountains etc., which are not easily accessible round the clock by humans. Besides, the traffic load of the nodes vary as each node can act as a router to forward the data to the sink. In order to estimate the traffic load at each node, in this section, we describe the system model and traffic load model.

3.1 System model

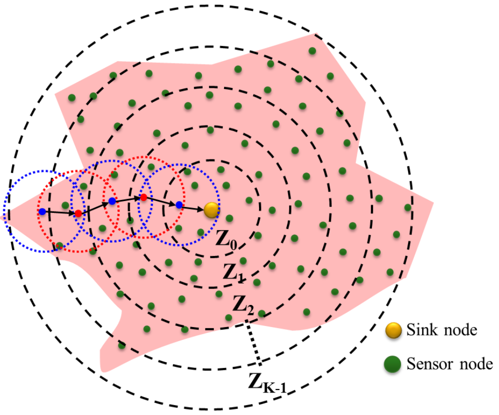
Let us assume that there are n numbers of sensor nodes $\{N_1, N_2, ..., N_n\}$, which are deployed randomly over an irregular monitoring region $\mathbb{R}$ as shown in Figure 1. In addition to n nodes, one special node known as Sink SN is also deployed to aggregate and transmit the data. Depending on the application requirements, SN may be located at inside or outside of the monitoring region. In this paper, it is assumed that SN is located at the centre of the monitoring region $\mathbb{R}$. In the post-deployment scenario, each node $N_i,\ \forall i = 1, 2, ..., n$ collects the data depending on the sensing range R_{S_i} and forward them to SN based on the communication range R_{C_i} in a single-hop or multi-hop fashion. In this paper, it is assumed that all sensors are homogeneous, which implies that for sensors N_i and N_j, $i \neq j$, $R_{S_i} = R_{S_j} = R_S$ and $R_{C_i} = R_{C_j} = R_C$. Further, it is also assumed that for each sensor, its communication range is equal to its sensing range, i.e., $R_C = R_S$. Besides, for simplicity, it is assumed that each sensor has same level of power at the time of deployment and has no mobility.

The entire monitoring region $\mathbb{R}$ is divided into set of K virtual zones $\mathbb{Z} = \{Z_0, Z_1, ..., Z_{K-1}\}$ around the SN and nodes are grouped into various zones based on their locations. Figure 1 shows an example of K virtual zones centred around SN. The zone formation process is described in detail in Section 4.1.1.

3.2 Traffic load model

In WSNs, random deployment of sensors results in an uneven density of sensors across the monitoring region. The areas densely covered by sensors are expected to generate more traffic; whereas, areas sparsely covered by sensors are expected to generate less traffic. Further, for those WSNs, where SN is deployed at the centre of the region, the amount of data traffic increases from outer zones towards the inner zones. Besides, when data from dense outer zones are forwarded to SN, sensors located in between them experience more traffic since they need to handle forwarded data in addition to their own. For these reasons, the data traffic generated across the region varies significantly from one zone to another, which greatly influences the power consumption of nodes. Hence, it is highly important to know the traffic load of each node for better power management. In this section, we describe the proposed traffic load model to estimate the amount of traffic for each node. Subsequently, the proposed traffic load model is used to design an efficient node scheduling algorithm for power management as described in Section 4.3.

Figure 1 Sensors deployment area with virtual zones (see online version for colours)

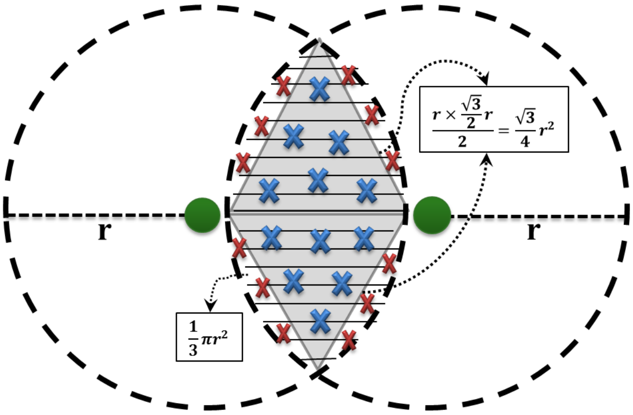


Usually, for simplicity, it is assumed that the nodes residing in the same zone are expected to have the same traffic load. However, in real circumstances, this assumption does not hold true. Let us assume that β_i represents the traffic load of node N_i, $\forall i = 1, 2, ..., n$. To design a better representative traffic load model, we first estimate the value of β_i using distance, zone area and density of the nodes within zone area. The distance between sensor node N_i to SN is measured in terms of the hop counts, i.e., χ_i, $\forall i = 1, 2, ..., n$. The χ_i can be calculated by the zone number labelled to node N_i. Secondly, the zone area is found using communication range, since zones are formed using communication range. Finally, the density of the zone area is calculated, which is represented as $D(\chi_i)$. The density is calculated by finding the number of neighbours in zone area and effective common communication range (ECCR). Figure 2 shows an example of ECCR between two sensors and the calculated area. Even though each sensor does not know its own location, it still can calculate the traffic load β_i adaptively using the mentioned procedure. The calculation of β_i is as shown in equation (1).

$$\beta_i = \frac{\lambda}{D(\chi_i)}\left(1 + \frac{1}{2(\chi_i)r(i)}(1 - \chi_i^2)\right), \tag{1}$$

where, χ_i is the distance from the sink node SN to the sensor node N_i, $D(\chi_i)$ is the density $(nodes/m^2)$ at χ_i, $\lambda/D(\chi_i)$ is the data rate transmitted by each node in the area and $r(i)$ is width of the ith zone.

Figure 2 The area of effective common communication range (see online version for colours)



4 Traffic load based scheduling algorithm

The proposed method is comprised of three parts i.e., zone formation, routing path selection algorithm and traffic load based node scheduling. The algorithms are explained and the relation of these parts will be established as follows.

4.1 *Zone formation and routing path selection*

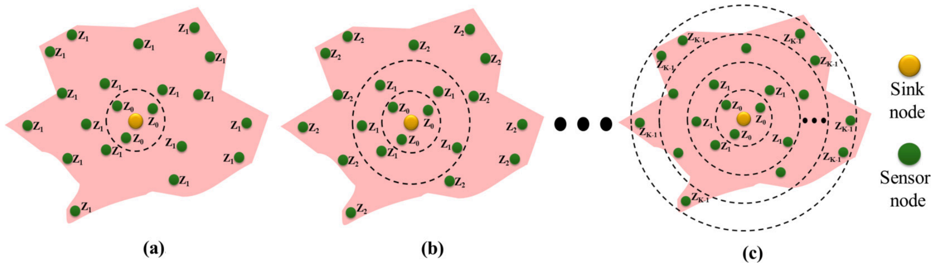
In this section, we will describe the process of zone formation and routing path selection after deployment of the nodes over an irregular monitoring region.

4.1.1 *Zone formation*

Once the sensor nodes are deployed randomly over an irregular monitoring region, we divide the entire region $\mathbb{R}$ into several zones for efficient coordination and data forwarding among sensor nodes. In this paper, it is assumed that the zone formation process initiates from the Sink node SN and continues until all the sensors are assigned to one of the zones covering the whole monitoring region. The total number of zones primarily depends on the area of the monitoring region and communication range R_C. At first, SN broadcasts a beacon packet with communication range R_C. The nodes within R_C of SN receive a beacon packet and send an acknowledgement packet to SN, which are assigned to Zone 0. Now let us assume that the entire monitoring region $\mathbb{R}$ is divided into K number of zones starting from Z_0, Z_1 up to Z_{K-1}. The successive zones are formed using the following property. The line of sight distance from a sensor residing in zone Z_i to sensor residing in zone Z_j, where $j = i + 1$ and $j \leq K - 1$ must be at most communication range R_C as shown in Figure 1.

The entire step-wise zone formation process is explained as shown Figure 3. Here, we have applied the incremental zone formation process, where zones are formed one after another based on the communication range R_C. As shown

Figure 3 Step-wise explanation of zones formation: (a) zone: Z_0; (b) zones: Z_0, Z_1 and (c) zones: $Z_0, Z_1, \ldots, Z_{K-1}$ (see online version for colours)



in Figure 3(a), all nodes within the communication range R_C of Sink node SN are labelled with zone Z_0 and rest are labelled with zone Z_1 irrespective of their location. Next, among the nodes labelled with zone Z_1, those lying outside the communication range R_C from already formed zone Z_0 are labelled with zone Z_2 and others are continued to label as zone Z_1 as shown in Figure 3(b). This process is repeated until all nodes are labelled with at least one zone number. Figure 3(c) shows the formation of K number of zones. The procedure of the above-mentioned zone formation mechanism is shown in Algorithm 1.

Algorithm 1: Procedure of Virtual Zone Formation.

```
Input:
    Sensor nodes $\{N_1, N_2, ..., N_n\}$,
    Sensing Range $R_S$,
    Communication Range $R_C$.
Output:
    Set of virtual zones $\mathbb{Z} = \{Z_0, Z_1, ..., Z_{K-1}\}$
1 Start from the Sink Node $SN$ ;
2 Check all the nodes within $R_C$ ;
3 Label nodes within $R_C$ to zone $Z_0$ ;
4 Starting from zone $Z_i$, where $i = 0, 1, ..., K-1$ ;
5 Check all the nodes within $R_C$ ;
6 Label nodes within $R_C$ to zone $Z_i$ ;
7 Label nodes outside $R_C$ with zone $Z_{i+1}$ ;
8 Repeat from Step 4 until all nodes are labeled ;
```

4.1.2 Routing path selection

The routing path selection is an important aspect of WSNs, which impacts the power consumption of the sensor nodes and thereby affects the network lifetime. In WSNs, based on the location of the nodes and surrounding node density, each node spends different amount of power for various activities such as listening, transmitting and receiving etc. For these reasons, nodes may have a different amount of residual power at any given time. Routing of data frames with unaware of residual power of the nodes may shorten the lifetime of the network. Hence, to improve the lifetime of the WSNs, we design a power aware routing path selection algorithm considering the location, i.e., zone number and residual power P_r of the nodes. The proposed path selection algorithm selects the neighbour node, which not only has the maximum amount of residual power but also lies in the immediate zone in decreasing order.

Let us assume that node $N_i, \forall i = 1, 2, ..., n$ has data for SN to send at time t. If node N_i is a sender that belongs to the zone Z_0 as per the zone formation process described earlier, it will transmit the data frame directly to the Sink SN through the single hop. However, if sender node N_i is belonged to zone Z_j other than $Z_0, \forall j = 1, 2, ..., K-1$, the data frame transmission to SN can be accomplished by sending the data frame to the neighbour nodes through the multi-hops. For a sender that belongs to the zone $Z_j, \forall j = 1, 2, ..., K-1$, it will first prepare a set Nbr of neighbour nodes within its communication range R_C residing in zone Z_{j-1}. Later, sender N_i sends a power query request packet Q_{Preq} to all the neighbour nodes to gather information about the remaining power of each neighbour node. Upon receiving Q_{Preq}, all neighbour nodes respond with Q_{Prly} packet indicating the respective remaining power P_r. Based on the Q_{Prly} packets, Sender N_i selects the node N_{sel} with maximum residual power and sends the data frame.

The mentioned path selection procedure is explained in detail with example as shown in Figure 4. Let us assume that the sender A from zone Z_3 wants to send a data frame to SN. Firstly, A broadcasts Q_{Preq} packet to neighbour nodes B, C and D residing in zone Z_2 and forwards the data frame to node C having maximum P_r as shown in Figure 4(a). The similar process is repeated by C, which forwards the data frame to E based on the maximum P_r criteria as shown in Figure 4(b). Finally, E forwards the data frame to I, which in turn forwards it to the Sink node SN as shown in Figure 4(c) and (d), respectively. The pseudo code for Routing Path Selection is given in Algorithm 2.

As the proposed scheduling protocol will be adjusted by the traffic load, the zone formation algorithm is needed. Each sensor may have different neighbours and the density in different zones. Hence, it will use the Effective Require Common Range to cope with the traffic load model that the variable traffic load can be found, then the result will be used in the quorum theorem to decide the quorum size and the scheduling protocol.

4.2 Quorum theory

In recent years, quorum theory is being considered as a powerful tool to apply in a wide range of applications such as distributed systems, wireless MANETs (Chang et al., 2016), cognitive radio networks (CRNs) (Sheu et al., 2016) and WSNs etc. In the context of WSNs, quorum theory is being used to design an energy-efficient node scheduling mechanism with the goal of prolonging the network lifetime by increasing the amount of sleep duration. The $G \times G$ quorum grid generates a cycle of G^2 number of equal length time frames, where each frame is consist of ad hoc traffic indication message (ATIM) window and data window. Depending on the chosen row and column, a sleep/wake up schedule of a node is prepared. An example 3×3 quorum grid is shown in Figure 5(a) for a random node B. Based on the random selection of row and column, corresponding schedule of nine frames is prepared as shown in Figure 5(b), where node B has to wake up in the frames numbered as 1, 4, 5, 6 and 7 during the ATIM window to check the pending data, whereas B can continue to sleep in the rest time frames.

Algorithm 2: Procedure of Routing Path Selection.

```
Input:
Sensor nodes $\{N_1, N_2, ..., N_n\}$,
Sensing Range $R_S$,
Communication Range $R_C$,
Remaining Power $P_r$.
Output:
Routing Path Set $\Psi$
Initialize Routing Path Set $\Psi = NULL$ ;
For all nodes $N_i$, where $i = 1, 2, ..., n$ ;
if $N_i$ has data then
    if $N_i \in Z_0$ then
        Transmit data directly to $SN$
    end
    if $N_i \notin Z_0$ then
        j = Retrieve Zone number of $N_i$ ;
        Set Current Zone to $Z_j$ ;
        while $Z_j > Z_0$ do
            Prepare Set $Nbr$ of neighbour nodes lying in Zone $Z_{j-1}$ ;
            $N_{sel}$ = Selected node from $Nbr$ having maximum $P_r$ ;
            Transmit data from $N_i$ to $N_{sel}$ ;
            Include $N_{sel}$ to Set $\Psi$ : $\Psi = \Psi \cup N_{sel}$ ;
            j = j - 1 ;
        end
        Transmit data directly to $SN$ ;
    end
    Routing Path Set $\Psi$ ;
end
```

Such predefined node scheduling eliminates the compulsory wake up in every frame (IEEE 802.11 Standard, 1999) and significantly reduces the power consumption during idle listening. However, grid-quorum based node scheduling is the static scheduling and it increases either the power consumption or latency if the grid size G is decided considering highly loaded node or lightly loaded node, respectively. For example, in $G \times G$ grid-based quorum, each sensor has the ratio of $\frac{(2G-1)}{G^2}$ wake up time. Hence, the grid size would influence the ratio of the wake up time and the sleeping time that the larger grid size has the lesser wake up time. Therefore, this paper proposes a traffic load aware node scheduling by dynamically adjusting the grid size per node that means the sleep/wake up scheduling to cope with varying traffic load without losing network connectivity.

Figure 4 Example of path selection procedure from zones: (a) Z3 to Z2; (b) Z2 to Z1; (c) Z1 to Z0 and (d) Z0 to Sink (see online version for colours)

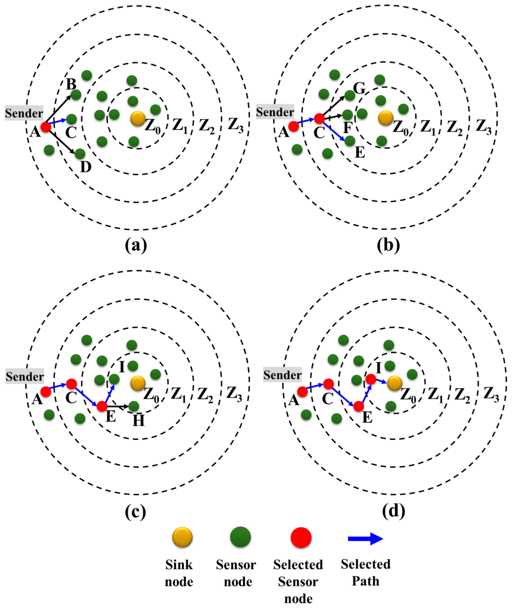
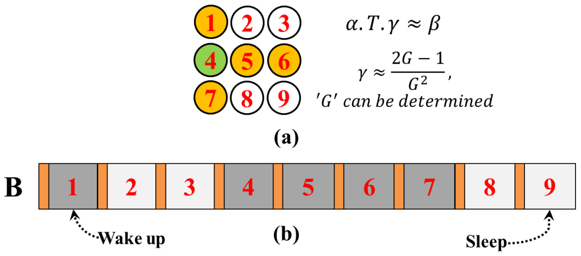


Figure 5 Example of: (a) grid-based quorum with $G = 3$ and (b) the corresponding node scheduling (see online version for colours)



4.3 Traffic load based scheduling (TLS)

In most quorum-based MAC protocols, simulation results are considered as a base to find out the traffic load and later the simulation based traffic loads are used to design the node scheduling algorithm, i.e., the scheduling of sleep/wake up time frames of sensors as well as to decide the quorum grid size. However, in real circumstances, the traffic load of nodes can change as time goes by. Considering the impact of varying traffic load on the scheduling of nodes, this paper proposes traffic load based scheduling (TLS) protocol, combining the traffic load model with quorum theory. Further, the proposed TLS is expected to adaptively adjust the quorum grid size. Since, each node has the same cycle time length T and α as the maximum data rate as assumed in this paper, each sensor can know the maximum traffic it can transmit in T, i.e., $\alpha.T$. It could be possible that the data rate may not always remain maximum and hence the node also does not need to wake up in all the time frames during the cycle and can continue to sleep to save the power. To ensure the increased sleep duration of nodes, TLS adjusts the scheduling protocol as follows. Each sensor relates the maximum data rate of a cycle with its own traffic load β_i, which is derived using traffic load model and calculates the Traffic Load Ratio i.e., γ_i. Since, in a $G \times G$ grid, sensor has the ratio of $\frac{(2G-1)}{G^2}$ wake up time frames to receive and transmit the data in cycle length T, the ratio γ_i can be equated with $\frac{(2G-1)}{G^2}$ to derive an approximation of G to set the initial quorum grid size. Finally, based on the varying traffic load β_i derived from the traffic load model, each sensor adjusts its quorum grid size dynamically and can efficiently consume its energy. The entire node scheduling procedure is explained step by step in Algorithm 3.

```
Algorithm 3: Traffic Load aware node scheduling.
  Input:
  $G \times G$: Inital Quorum grid size and $G \geq 1$,
  $\alpha$: Maximum data rate of each node,
  $\beta_i$: Traffic load of node $N_i$ from Traffic load Model,
  $T$: Cycle time of each sensor.
  Output:
  $\widehat{G} \times \widehat{G}$: New Quorum Grid Size and $\widehat{G} \geq 1$.
  Notations:
          $\gamma_i$: Traffic Load ratio of node $N_i$.
  Initialize Quorum grid size for each node $N_i$,
    $\forall i = 1, 2, ..., n$;
  Obtain maximum traffic load of each node in a
    cycle: $\alpha T$;
  Calculate Traffic Load ratio of each node:
    $\gamma_i = \frac{\beta_i}{\alpha T}$;
  Relate $\gamma_i$ with quorum wake up frequency of
    $G \times G$ grid : $\gamma_i = \frac{2G-1}{G^2}$ ;
  Simplify to quadratic function:
    $G^2.\gamma_i - 2G + 1 = 0$, where $\gamma_i$ is constant. ;
  Find the roots and obtain $\widehat{G} \geq 1$ ;
```

The primary concern of having different quorum grid size of nodes is the network connectivity. In other words, different quorum grid size will lead to have a different number of time frames in a same cycle length T with different sleep/wake up time frames, which may prevent any two nodes to meet in the same time frame. However, as shown in Figure 6(b), the example shows sensors can communicate when quorum sizes are different. Figure 6(a) shows three different quorum grid sizes, i.e., 2×2, 3×3 and 4×4 for nodes A, B and C, respectively and Figure 6(b) shows the corresponding sleep/wake up schedules. From Figure 6(b), it is clear that nodes A, B and C meet successfully in multiple common time frames with cycle length T though they have different quorum size.

4.4 Latency analysis

In this section, we present the theoretical analysis of latency based on different modes of the sensors such as active or sleep.

4.5 Latency

In our analysis, the latency (Δ) is defined as the delay in delivering the received data traffic and therefore incurring the undesired overhead in different power saving modes.

$$\Delta = \left[\frac{\sum_{i=1}^{M}(\xi_i - \varepsilon_i)}{M}\right], \tag{2}$$

where

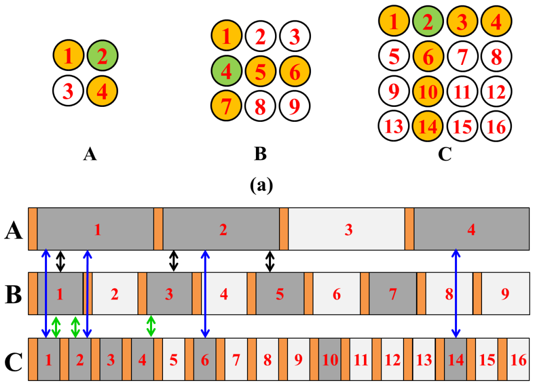
M: number of data frames delivered

ξ_i: delivery time of ith data frame

ε_i: arrival time of ith data frame.

Here, we define three different types of latencies, i.e., sleep (Δ_S), idle (Δ_I) and dormant (Δ_D) based on the modes. It is observed that for any power conserving method, its efficiency relies on the duration of the sleep/idle window. The longer the duration of the sleep/idle window, more efficient the method can be. On the contrary, longer duration of the sleep/idle window also results with longer transmission time, i.e., latency. Hence, efficiency and latency are always orthogonal to each other. Equations given below give the basis of our latency calculation scheme.

Figure 6 Example TLS execution for nodes A, B and C with: (a) grid size 2 × 2, 3 × 3 and 4 × 4; and (b) corresponding scheduling of nodes (see online version for colours)



(a)

(b)

4.5.1 Sleep latency Δ_S

To determine the value of Δ_S, values of data frame delivery time ξ_i and arrival time ε_i are required. We know that, the arrival time of traffic addressed to a node for the ith data frame is same as the time of completion of inactivity period of the node i.e., $\varepsilon_i = tc_i$. For calculation of ε_i, value of an integer '**a**', is determined such that,

$$(s_{\text{int}} + (a-1)s_{\text{inc}}) \leq s_{\text{max}} < (s_{\text{int}} + a.s_{\text{inc}}), \tag{3}$$

where

s_{max}: allowed maximum length of sleep window

s_{inc}: length by which sleep window is augmented at a time

s_{int}: initial length of sleep window.

Note that value of '**a**' remains same for all data frame delivery and gives the number of times sleep window can be augmented before achieving maximum value. After the determination of value of '**a**', tc_i is compared against time for which length of sleep window augments and is given by,

$$\left(a.\left(s_{\text{int}} + l\right) + a.\left(\frac{(a-1)}{2}\right)s_{\text{inc}}\right). \tag{4}$$

- *Case I* If $tc_i \leq \left(a\left(s_{\text{int}}+l\right)+a\left(\frac{(a-1)}{2}\right)s_{\text{inc}}\right)$.

This case represents the scenario, where traffic arrives while sleep windows are still augmenting the integral value of a parameter m_i as given below

$$tc_i \leq \left(m_i(l+s_{\text{int}})+m_i\left(\frac{(m_i-1)}{2}\right)s_{\text{inc}}\right) \tag{5}$$

$$tc_i > (m_i-1)(l+s_{\text{int}})+(m_i-1)\left(\frac{(m_i-2)}{2}\right)s_{\text{inc}} \tag{6}$$

m_i gives number of the interval at which node receives the traffic and is used to compute the start of the interval, x_i.

$$x_i = (m_i-1)(l+s_{\text{int}})+(m_i-1)\left(\frac{(m_i-2)}{2}\right)s_{\text{inc}}. \tag{7}$$

With the help of x_i and ε_i, offset_i is calculated as

$$\text{offset}_i = (\varepsilon_i - x_i) = (tc_i - x_i). \tag{8}$$

offset_i is then used to decide whether the traffic arrived during listening or sleeping interval of a node. Arrival of traffic during listening period of a node is represented by the condition,

$$\text{offset}_i \leq m_i\left(s_{\text{int}}+\frac{(m_i-1)}{2}s_{\text{inc}}\right). \tag{9}$$

In this case, sleep latency for ith data frame remains zero, i.e. $\Delta_{S_i}=0$. On the other hand, the situation when traffic arrives while node is sleeping is represented by the condition,

$$\text{offset}_i > m_i\left(s_{\text{int}}+\frac{(m_i-1)}{2}s_{\text{inc}}\right). \tag{10}$$

Accordingly, sleep latency for this case is given by,

$$\Delta_{S_i} = \left(m_i(l+s_{\text{int}})+m_i\left(\frac{m_i-1}{2}\right)s_{\text{inc}}\right)-tc_i. \tag{11}$$

- *Case II* If $tc_i \geq \left(a(s_{\text{int}}+l)+a\left(\frac{(a-1)}{2}\right)s_{\text{inc}}\right)$

This case holds good when traffic targeted to a node arrives when sleep window has already attained its maximum value. The calculations for latency in this case are minimised by chopping off the part of sleep mode in which sleep window is augmented. Doing so is acceptable as that part has got no role to play in latency determination in this case.

$$tc_{i_{\text{new}}} = tc_i - \left(a(s_{\text{int}}+l)\right)+a\left(\frac{a-1}{2}\right)s_{\text{inc}}. \tag{12}$$

$tc_{i_{\text{new}}}$ is then used for the computation of $offset_i$, which is a measure of time difference in arrival of traffic and end of the previous sleep period.

$$\text{offset}_i = tc_{i_{\text{new}}} - \left\lfloor\frac{tc_{i_{\text{new}}}}{l+s_{\max}}\right\rfloor(l+s_{\max}). \tag{13}$$

offset_i helps in determining whether the traffic arrived during the sleep period or listening period and thus compute the sleep latency accordingly.

If, $\text{offset}_i \leq s_{\max}$, then Δ_{S_i} is set to zero. As this condition represents the arrival of traffic while node is listening. On the other hand, $offset_i > s_{\max}$ represents the scenario, where traffic arrives while node is sleeping. Under this condition, Δ_{S_i} is determined as follows.

$$\Delta_{S_i} = ((l+s_{\max})-\text{offset}_i). \tag{14}$$

4.5.2 *Idle latency Δ_I*

Δ_I is calculated by first calculating the Δ_{I_i} for each data frame and then averaging it over the number of data frames. As stated before, the arrival time of traffic addressed to a node in ith interval is same as the time of completion of inactivity period of that node, i.e., $\varepsilon_i = tc_i$. Hence, the value of ε_i is only required for the calculation of Δ_{I_i} and is computed by first calculating the $excess_i$, which can be derived as follows.

$$\text{excess}_i = \left[tc_i - \left\lfloor\frac{tc_i}{(pa+pu)}\right\rfloor(pa+pu)\right], \tag{15}$$

where, pa is sensing available period and pu is sensing unavailable period. Value of $excess_i$ is then used to determine whether the traffic arrived during sensing available or unavailable period of a node.

- *Case I*: If $\text{excess}_i \leq pa$ This case depicts the condition when traffic arrives in the sensing available period of a node and hence, Δ_{I_i}.
- *Case II*: If $\text{excess}_i > pa$ This represents the scenario where traffic arrives during sensing unavailable period. Hence, $\Delta_{I_i} = ((pa+pu)-\text{excess}_i)$.

4.5.3 *Dormant latency Δ_D*

In our analysis, the dormant mode is defined as the combination of sleep and idle modes and accordingly, dormant latency is calculated as given below.

If $tc_i \leq T_S$, then $\Delta_{T_i} = \Delta_{S_i}$

If $tc_i > T_S$, then $\Delta_{T_i} = \Delta_{I_i}$.

5 Performance evaluation

In this section, we analyse the performance of our zone formation algorithm and TLS with AQEC (Chao et al., 2006) and QMAC_LR (Chao and Lee, 2010) algorithms through simulation. The detail description of the simulation setups and results are given as follows.

5.1 *Simulation setups*

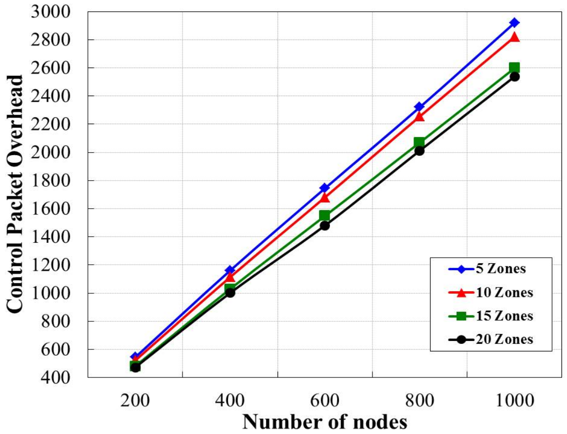
In our simulation, a monitoring region of size 300 × 300 m^2 is considered. The numbers of deployed nodes over the monitoring region are taken to be 200, 400, 600, 800 and 1000 nodes. It is assumed that each node can generate 100 bytes of packet in every 5 s and the channel capacity is considered to be 10 kb/s. The initial energy of each sensor is 100 J, and the energy consumption for each transmission is 0.5 J. We use Dev-C++ 4.9.9.2 to evaluate the performance of our proposed method.

5.2 Simulation results

In this section, it shows two parts of results. The first part, it calculates the control packet overhead on Zone formation Algorithm in different situations, such as sensors set the zone a with different number of nodes and the different zones with the same nodes. The second part regards the total bandwidth consumption when the sensor generates the packet and transmits it from the last zone to the sink. It also discusses this result both constant bit rate (CBR) and variable bit rate (VBR). These results will compare with AQEC and QMAC_LR.

As shown in Figure 7, control packet overhead for setting different number of zones with a different numbers of nodes. In this simulation, it can be divided into two parts to explain. The control packet overhead grows up when the number of nodes increases. The other one is different number of nodes in the same zone. In our zone formation algorithm, the sensor sets zone and routing path hop-by-hop and transmits packet zone-by-zone. It implies that the same number of nodes with fewer zones must communicate more times either in the same zone or in the next zone. For this reason, control packet overhead is inversely proportional to the number of zones.

Figure 7 Control packet overhead with different number of zones (see online version for colours)



As shown in Figure 8, the number of zones is fixed at 15 and the communication range of the nodes is changed to evaluate the control packet overhead. It is observed that the longer communication range can communicate with more nodes for which the received packet overhead is increasing. Figure 9 shows the energy consumption when nodes are distributed over different zones with different numbers of nodes. The energy consumption is related to the control packet overhead. The higher the overhead is, the more energy is consumed. The figure is presented by the ratio of the total nodes. When the number of nodes is more than or equal to 800, the ratio of energy consumption is approximately equal to the same.

The last of this part, Figure 10 shows control packet overhead comparisons with other two algorithms, AQEC and QMAC_LR. This simulation also fixes 15 zones. AQEC has the largest control packet overhead; each sensor just broadcasts packets and receives packets except for out of communication range, therefore the system overhead is large. The second is QMAC_LR that it uses next hop group member to set which nodes can receive the packet in the inner zone. In our algorithm, the zone setting is hop-by-hop, and each node chooses the one which has the highest energy in the inner zone to transmit the packet. It is clear that our algorithm acquired the better result than them.

Figure 8 Control packet overhead with variable communication range (see online version for colours)

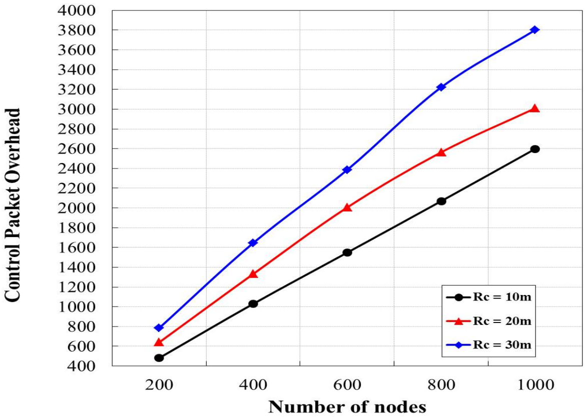
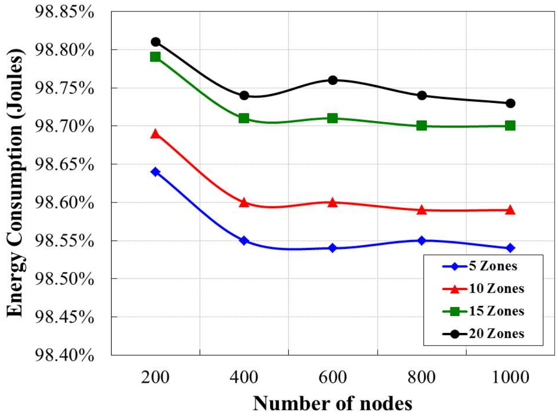


Figure 9 Energy consumption for different number of zones (see online version for colours)



The result of bandwidth consumption from the last zone to the sink is depicted as shown in Figure 11(a) for CBR and Figure 11(b) for VBR, respectively to confirm the proposed protocol can adjust the quorum size to cope with the different traffic load efficiently. The sensor node in the last zone will generate data packets and transmit to the inner zone hop-by-hop in the 100 s simulation time. These figures show the average of total bandwidth consumption in each zone except for the sink node. In Figure 11(a), CBR is constant bit rate that proposed protocol can adjust the quorum size by the fix bit rate when the first round is terminated. For this reason, the analysis results almost have no waste at time frames. In Figure 11(b), it uses the traffic load model to cope with different sensor density in different zones as mentioned previously.

Figure 10 Comparisons of control packet overhead with fixed numbers of zones (see online version for colours)

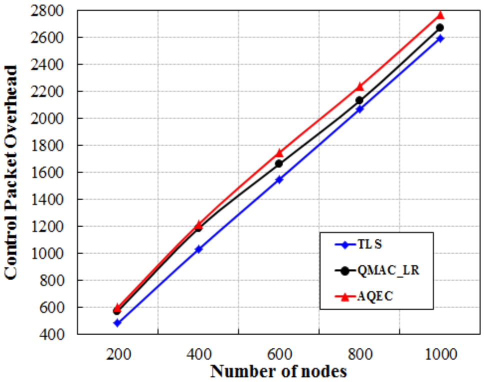
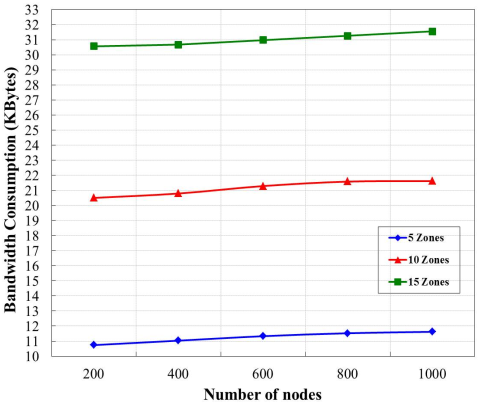
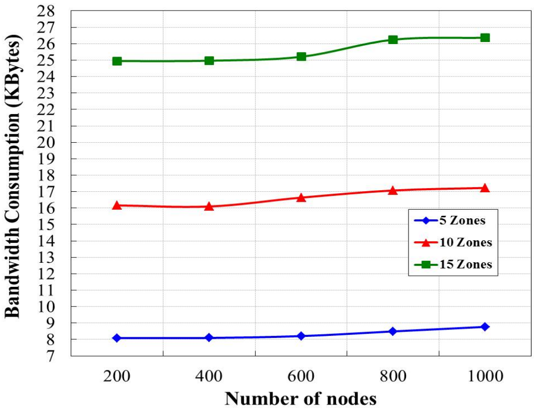


Figure 11 Bandwidth consumption for transmitting data packets from the last zone to the sink with CBR (a) and VBR (b) (see online version for colours)



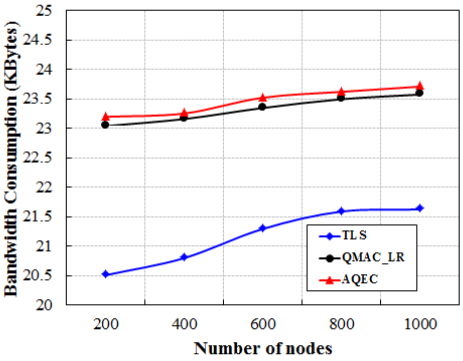
(a)



(b)

In Figure 12, we compare the bandwidth consumption with AQEC and QMAC_LR in CBR. In our proposed protocol TLS, it is effective to adjust the quorum size for suitable load expect for a large number of nodes will cause the heavier load. AQEC and QMAC_LR use the fix quorum size that decides by the different corona. Therefore, when the traffic is light that the node has large quorum size still wake up and it will waste a lot of energy. On the other hand, it will cause a lot of delays when the small quorum size confronts the heavy load.

Figure 12 Comparison of bandwidth consumption for transmitting data packets from the last zone to the sink with CBR (see online version for colours)



6 Conclusion

In this paper, a zone formation and routing path selection algorithm is proposed first for setting different zones, which can decide the nodes with high or low residual energy level to receive the data packets from the preceding zone to the sink. The result shows that control packet overhead and energy consumption are reduced considerably. The traffic load based scheduling (TLS) protocol is designed by combining the traffic load model with the quorum theory. In TLS, each sensor can calculate its own total traffic load based on the density of nodes in each zone, its communication range and hop counts from the sink node. Hence, a node can easily estimate the number of time frames it can wake up. Based on the concerned traffic load and quorum, the suitable grid size can be adjusted, which implies that the sleep/wake up scheduling can be controlled efficiently. Thus, the sensor network can cope with unexpectedly events easily and is therefore more suitable for applications where variable traffic load is observed.

Acknowledgements

This work is supported by the Ministry of Science and Technology (MOST), Taiwan under Grant number 107-2221-E-182-073.

References

Agrawal, D. and El Abbadi, A. (1991) 'An efficient and fault-tolerant solution for distributed mutual exclusion', *ACM Transactions on Computer Systems (TOCS)*, Vol. 9, No. 1, pp.1–20.

Anisi, M.H., Abdullah, A.H., Coulibaly, Y. and Razak, S.A. (2013) 'EDR: efficient data routing in wireless sensor networks', *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 12, No. 1, pp.46–55.

Chang, C-T., Chang, C-Y., Kuo, C-H. and Hsiao, C-Y. (2016) 'A location-aware power saving mechanism based on quorum systems for multi-hop mobile ad hoc networks', *Ad Hoc Networks*, Vol. 53, pp.94–109.

Chao, C-M. and Lee, Y-W. (2010) 'A quorum-based energy-saving mac protocol design for wireless sensor networks', *IEEE Transactions on Vehicular Technology*, Vol. 59, No. 2, pp.813–822.

Chao, C-M., Lien, L-F. and Hsu, C-Y. (2013) 'Rendezvous enhancement in arbitrary-duty-cycled wireless sensor networks', *IEEE Transactions on Wireless Communications*, Vol. 12, No. 8, pp.4080–4091.

Chao, C-M., Sheu, J-P. and Chou, I-C. (2006) 'An adaptive quorum-based energy conserving protocol for IEEE 802.11 ad hoc networks', *IEEE Transactions on Mobile Computing*, Vol. 5, No. 5, pp.560–570.

Chen, Y-S., Chen, H-Z. and Cheng, R-S. (2015) 'Adaptive quorum-based power-saving protocol for asynchronous wireless ad hoc networks', *Wireless Personal Communications*, Vol. 84, No. 3, pp.1771–1795.

Cheung, S.Y., Ammar, M.H. and Ahamad, M. (1992) 'The grid protocol: a high performance scheme for maintaining replicated data', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 4, No. 6, pp.582–592.

Chou, Z-T., Lin, Y-H. and Jan, R-H. (2011) 'Optimal asymmetric and maximized adaptive power management protocols for clustered ad hoc wireless networks', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 22, No. 12, pp.1961–1968.

Chou, Z-T., Lin, Y-H. and Sheu, T-L. (2013) 'Asynchronous power management protocols with minimum duty cycle and maximum adaptiveness for multihop ad hoc networks', *IEEE Transactions on vehicular technology*, Vol. 62, No. 7, pp.3301–3314.

Ha, R.W., Ho, P-H. and Shen, X. (2006) 'Optimal sleep scheduling with transmission range assignment in application-specific wireless sensor networks', *International Journal of Sensor Networks*, Vol. 1, Nos. 1-2, pp.72–88.

Heinzelman, W., Ba, H. and Chen, L. (2016) *Wireless Sensor Network Wake-up Range Extension via Energy Harvesting and Edge Devices*, US Patent 9,232,475.

Heinzelman, W.R., Kulik, J. and Balakrishnan, H. (1999) 'Adaptive protocols for information dissemination in wireless sensor networks', *Proceedings of the 5th annual ACM/IEEE International Conference on Mobile Computing and Networking*, ACM, Seattle, Washington, USA, pp.174–185.

Hwang, R-H., Wang, C-Y., Wu, C-J. and Chen, G-N. (2013) 'A novel efficient power-saving mac protocol for multi-hop MANETs', *International Journal of Communication Systems*, Vol. 26, No. 1, pp.34–55.

IEEE 802.11 Standard (IEEE Computer Society LAN MAN Standards Committee).

Malekshan, K.R., Zhuang, W. and Lostanlen, Y. (2014) 'An energy efficient MAC protocol for fully connected wireless ad hoc networks', *IEEE Transactions on Wireless Communications*, Vol. 13, No. 10, pp.5729–5740.

Olariu, S. and Stojmenovic, I. (2005) 'Data-centric protocols for wireless sensor networks', *Handbook of Sensor Networks: Algorithms and Architectures*, pp.417–456.

Peng, M., Xiao, Y. and Sun, B. (2017) 'Utilisation and energy consumption of fair-access in sensor networks with selected grid topologies', *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 24, Nos. 1–2, pp.117–133.

Rault, T., Bouabdallah, A. and Challal, Y. (2014) 'Energy efficiency in wireless sensor networks: A top-down survey', *Computer Networks*, Vol. 67, pp.104–122.

Sahoo, P.K., Hwang, I-S. and Cheng, C-Y. (2012) 'An adaptive traffic load based scheduling protocol for wireless sensor networks', *2012 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, IEEE, pp.1–6.

Sherly Puspha, L. and Murugan, K. (2015) 'Energy-efficient quorum-based MAC protocol for wireless sensor networks', *ETRI Journal*, Vol. 37, No. 3, pp.480–490.

Sheu, J-P., Su, C-W. and Chang, G-Y. (2016) 'Asynchronous quorum-based blind rendezvous schemes for cognitive radio networks', *IEEE Transactions on Communications*, Vol. 64, No. 3, pp.918–930.

Sohrabi, K., Gao, J., Ailawadhi, V. and Pottie, G.J. (2000) 'Protocols for self-organization of a wireless sensor network', *IEEE Personal Communications*, Vol. 7, No. 5, pp.16–27.

Thomas, R.H. (1979) 'A majority consensus approach to concurrency control for multiple copy databases', *ACM Transactions on Database Systems (TODS)*, Vol. 4, No. 2, pp.180–209.

Van Dam, T. and Langendoen, K. (2003) 'An adaptive energy-efficient MAC protocol for wireless sensor networks', *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, Los Angeles, California, USA, ACM, pp.171–180.

Wu, S-H., Chen, C-M. and Chen, M-S. (2011) 'Collaborative wakeup in clustered ad hoc networks', *IEEE Journal on Selected Areas in Communications*, Vol. 29, No. 8, pp.1585–1594.

Wu, S-H., Chen, M-S. and Chen, C-M. (2014) 'Optimally adaptive power-saving protocols for ad hoc networks using the hyper quorum system', *IEEE/ACM Transactions on Networking*, Vol. 22, No. 1, pp.1–15.

Xi, Y., Jun, L. and Fang, Y. (2010) 'Variable grid-quorum pattern mac protocol for wireless sensor network', *2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering*, Vol. 1, IEEE, Changchun, China, pp.387–390.

Ye, W., Heidemann, J. and Estrin, D. (2004) 'Medium access control with coordinated adaptive sleeping for wireless sensor networks', *IEEE/ACM Transactions on networking*, Vol. 12, No. 3, pp.493–506.

Zheng, T., Radhakrishnan, S. and Sarangan, V. (2005) 'PMAC: an adaptive energy-efficient mac protocol for wireless sensor networks', *19th IEEE International Parallel and Distributed Processing Symposium*, IEEE, Denver, CO, USA, p.8.

Zytoune, O., Fakhri, Y. and Aboutajdine, D. (2010) 'Lifetime maximisation algorithm in wireless sensor network', *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 6, No. 3, pp.140–149.