iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** XII **Month of publication:** December 2017

DOI:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 5 Issue XII December 2017- Available at www.ijraset.com

AN Energy Efficient MAC Protocol for WSN based on Quorum Concept

M. Chandrasekar[1], T. Ashok[2]

***Abstract:** The lifetime of each node in a wireless sensor network depends on the durability of the wireless sensor nodes battery resources. Hence, the power consumption of sensor node must be tightly controlled. This can be done by designing energy saving MAC protocols. Most of the existing power-saving protocols achieve power savings by periodically putting sensor nodes to sleep. Such a regular sleep/awake mechanism fails to adjust a sensor node's sleep duration based on its traffic load, thereby reducing its power efficiency and increase the latency. In this paper, we propose a quorum-based medium access control (QMAC) protocol that enables sensor nodes to sleep longer under light loads. Since traffic flows toward the sink node in wireless sensor networks, a new concept, i.e., the next-hop group, is also proposed to reduce transmission latency. Simulation results verify that the proposed QMAC saves more energy and keeps the transmission latency low.*

I. INTRODUCTION

A. *Wireless Sensor Networks*

Wireless Sensor Networks consists of individual nodes that are able to interact with their environment by sensing or controlling physical parameter; these nodes have to collaborate in order to fulfill their tasks as usually, a single node is incapable of doing so; and they use wireless communication to enable this collaboration. The definition of WSN, according to, Smart Dust program of DARPA is: "A sensor network is a deployment of massive numbers of small, inexpensive, self powered devices that can sense, compute, and communicate with other devices for the purpose of gathering local information to make global decisions about a physical environment".

B. *Wireless Sensor Network Model*

Unlike their ancestor ad-hoc networks, WSNs are resource limited, they are deployed densely, they are prone to failures, the number of nodes in WSNs is several orders higher than that of ad hoc networks, WSN network topology is constantly changing, WSNs use broadcast communication mediums and finally sensor nodes don't have a global identification tags. The major components of a typical sensor network are:

1) *Sensor Field:* A sensor field can be considered as the area in which the nodes are placed.
2) *Sensor Nodes:* Sensors nodes are the heart of the network. They are in charge of collecting data and routing this information back to a sink.
3) *Sink:* A sink is a sensor node with the specific task of receiving, processing and storing data from the other sensor nodes. They serve to reduce the total number of messages that need to be sent, hence reducing the overall energy requirements of the network. Sinks are also known as data aggregation points.
4) *Task Manager:* The task manager also known as base station is a centralized point of control within the network, which extracts information from the network and disseminates control information back into the network. It also serves as a gateway to other networks, a powerful data processing and storage centre and an access point for a human interface. The base station is either a laptop or a workstation.

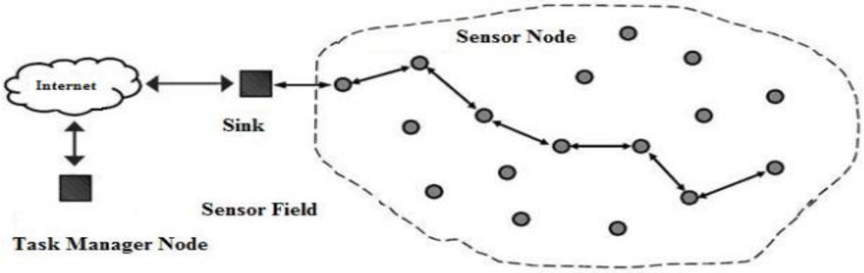

Fig.1.1. Components of Wireless Sensor Networks Data

C. The Sensor Node

A sensor is a small device that has a micro-sensor technology, low power signal processing, low power computation and a short-range communications capability. Sensor nodes are conventionally made up of four basic components a sensor, a processor, a radio transceiver and a power supply/battery. Additional components may include Analog-to-Digital Converter (ADC), location finding systems, mobilizes that are required to move the node in specific applications and power generators. The analog signals are measured by the sensors are digitized via an ADC and in turn fed into the processor. The processor and its associated memory commonly RAM is used to manage the procedures that make the sensor node carry out its assigned sensing and collaboration tasks. The radio transceiver connects the node with the network and serves as the communication medium of the node. Memories like EEPROM or flash are used to store the program code. The power supply/battery is the most important component of the sensor node because it implicitly determines the lifetime of the entire network. Due to size limitations of AA batteries or quartz, cells are used as the primary sources of power. To give an indication of the energy consumption involved, the average sensor node will expend approximately 4.8mA receiving a message, 12mA transmits a packet and 5µA sleeping. In addition the CPU uses on average 5.5mA when in active mode.

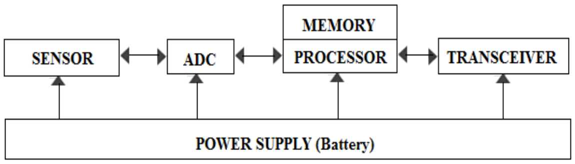

Fig.1.2. Components of a Wireless Sensor Node

D. Characteristics Of Wireless Sensor Networks

1) *WSNs have some unique characteristics. These are:*
   - *a)* Sensor nodes are small-scale devices with volumes approaching cubic millimeters in the near future. Such small devices are very limited in the amount of energy they can store or harvest from the environment.
   - *b)* Nodes are subject to failures due to depleted batteries or, more generally, due to environmental influences. Limited size and energy also typically means restricted resources (CPU performance, memory, wireless communication bandwidth and range).
   - *c)* Node mobility, node failures, and environmental obstructions cause a high degree of dynamics in WSN. This includes frequent network topology changes and network partitions. Despite partitions, however, mobile nodes can transport information across partitions by physically moving between them.
   - *d)* The resulting paths of information flow might have unbounded delays and are potentially unidirectional. Communication failures are also a typical problem of WSN.

E. Design Factors

1) *The following factors are important as a guideline to design a protocol for wireless sensor networks:*
   - *a) Fault tolerance*: The failure of some sensor nodes must not affect the overall task of the network.
   - *b) Scalability:* The protocol must be able to work in dense wireless sensor networks.
   - *c) Production costs:* The cost of producing a node must be kept low.
   - *d) Hardware constraints:* A sensor node is made up of four basic components: a sensing unit, a processing unit, a transceiver unit, and a power unit. Apart from size, there are some other stringent constraints for sensor nodes. These nodes must consume extremely low power, operate in high volumetric densities, have low production cost, be dispensable and autonomous, operate unattended, and be adaptive to the environment.
   - *e) Sensor network topology:* Wireless sensor network protocol must be able to maintain network topology so the network can accomplish its objectives.
   - *f) Power consumption:* In a multi-hop wireless sensor network, each node plays the dual role of data originator and data router. The malfunctioning of a few nodes can cause significant topological changes and might require rerouting of packets and reorganization of the network. Hence, power conservation and power management take on additional importance. Our protocol will focus on reducing network energy consumption and latency.

F. Power Saving In Wireless Sensor Network

Generally, the sensor nodes are battery powered, and it is often not feasible to recharge them. Thus, it is important to design energy-efficient protocols for WSNs. Recently, many protocols have been proposed to extend the network lifetime of sensor networks in their deployment protocols, power efficient medium access control (MAC) protocols and routing protocols.

In wireless sensor applications where all sensor nodes constantly report data to a single sink node (a many-to-one communication model), sensor nodes that are closer to the sink deplete their power faster. This is referred to as the energy-hole problem. To solve this, node deployment protocols try to distribute more nodes around the sink. However, due to environment limits, sometimes, only uniform (random) node distribution is possible. In such situations, node-deployment protocols are unable to prolong the network lifetime. One way of prolonging the network lifetime is by designing energy-efficient MAC protocols. Since idle listening has been identified as a major reason for energy wastage, several proposals were made to reduce the time a sensor node spends in idle listening. Some required time synchronization among sensor nodes. Since time synchronization is essential for many sensor applications, it is natural for a synchronous MAC protocol to be used to conserve energy. Generally, synchronous protocols maintain a schedule that indicates when a sensor should be awake to check transmission activity. Energy waste is reduced since nodes keep awake only at a specified time. However, these synchronous protocols either suffer from long latency or fail to adapt to an individual’s traffic well. In asynchronous solutions, sensor nodes independently schedule their awake period to periodically check the channel. In transmitting data, the source node sends a preamble that is long enough for the destination node to detect. When the preamble is detected, the destination node will remain awake to receive the data that follows the preamble. For asynchronous sensor applications, these protocols avoid the synchronization overhead. However, long preambles introduce long latency and excess energy consumption. All of the protocols aforementioned, including synchronous and asynchronous ones, do not consider the problem wherein nodes may unexpectedly fail. Unexpected node failures produce link breakage, which prevents data packets from being transmitted smoothly. As a result, the transmission latency is enlarged. The protocol we propose in this paper is also a synchronous MAC protocol. The core of this proposed protocol is a quorum-based wake-up scheme. The wake-up frequency is determined according to each sensor node’s traffic load. A node is allowed to sleep longer if less traffic is involved. The concept of quorum has been used to design energy conservation MAC protocols for wireless mobile ad hoc networks. Here, we investigate the effectiveness of using quorum in MAC protocol design for WSNs. Since latency is an important issue, we also provide a scheme that reduces the sensor-to-sink latency. This scheme proves to be robust in an environment where unexpected node failures exist.

Wireless sensor networks are dense networks of small, low-cost sensors, which collect and disseminate data. Wireless sensor networks facilitate monitoring and controlling of physical environments from remote locations. A sensor network is composed of a large number of sensor nodes that are densely deployed either inside the phenomenon or very close to it. The position of sensor nodes need not be engineered or predetermined. This allows random deployment in inaccessible terrains or disaster relief operations. On the other hand, this also means that sensor network protocols and algorithms must possess self-organizing capabilities. Another unique feature of sensor networks is the cooperative effort of sensor nodes. Sensor nodes are fitted with an onboard processor. Instead of sending the raw data to the nodes responsible for the fusion, they use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data.

G. Reasons For Energy Waste

The following reasons are main reason for energy waste.

1) *Collision:* When a transmitted packet is corrupted it has to be discarded, and the follow-on retransmissions increase energy consumption. Collision increases latency as well.
2) *Overhearing:* Overhearing, meaning that a node picks up packets that are destined to other nodes.
3) *Control packet overhead:* Sending and receiving control packets consumes energy too, and less useful data packets can be transmitted.
4) *Idle listening:* listening to receive possible traffic that is not sent. However idle listening is the major reason for energy waste. Many measurements have shown that idle listening consumes 50–100% of the energy.

H. Mac Protocols For Wireless Sensor Networks

The MAC is a sub layer of the Data Link Layer of the Open Systems Interconnection (OSI) model. Among other things, this layer is responsible for controlling the access of nodes to the medium to transmit or receive data. In traditional wireless voice or data networks, each user desires equal opportunity and time to access the medium, i.e., sending or receiving packets for their own applications. Per-hop MAC level fairness is, thus, an important issue. However, in sensor networks, all nodes cooperate for a single common task. At any particular time, one node may have dramatically more data to send than some other nodes. In this case fairness is not important as long as application-level performance is not degraded. In wireless sensor networks, MAC protocols control how sensor nodes access a shared radio channel to communicate with neighbors. Traditionally, this problem is known as the channel allocation or multiple access problems. Though MAC protocols have been extensively studied in traditional areas of wireless voice and data communications (e.g. Time division multiple access (TDMA), frequency division multiple access (FDMA), code division multiple access (CDMA), and carrier sense multiple access (CSMA)), sensor networks require a MAC protocol that differs from those of traditional wireless voice or data networks in several ways. First of all, most nodes in sensor networks are likely to be battery powered and it is often very difficult to change batteries for all the nodes. Second, nodes are often deployed in an ad-hoc fashion rather than with careful pre-planning. Hence, after deployment the sensor nodes must quickly organize themselves into a communication network. Third, many applications employ large numbers of nodes. Finally, most traffic in the network is triggered by sensing events, and it can be extremely burst. All these characteristics suggest that traditional MAC protocols employed in the past wireless networks are not suitable for wireless sensor networks without modifications.

Since our protocol is designed to detect extreme events, the system thus has to remain operational for months or years. Once a flooding is detected, this information must be forwarded to the system management quickly and accurately. Contention based MAC protocols are suitable for wireless sensor networks because the required synchronization allows higher clock drifts; while TDMA requires a tight synchronization. The lifetime of a mobile ad hoc network (MANET) depends on the durability of the mobile hosts' battery resources. In the IEEE 802.11 Power Saving Mode, a host must wake up at every beacon interval, to check if it should remain awake. Such a scheme fails to adjust a host's sleep duration according to its traffic, thereby reducing its power efficiency. This paper presents new MAC protocols for power saving in a single hop MANET. The essence of these protocols is a quorum-based sleep/wake-up mechanism, which conserves energy by allowing the host to sleep for more than one beacon interval, if few transmissions are involved. The proposed protocols are simple and energy-efficient. Simulation results showed that our protocols conserved more energy and Extended the lifetime of a MANET.

The T-MAC [4] is a contention-based Medium Access Control protocol for Wireless sensor networks. Applications for these networks have some characteristics (low message rate, insensitivity to latency) that can be exploited to reduce energy consumption. To handle load variations in time and location, T-MAC [4] introduces an adaptive duty cycle in a novel way: by dynamically ending the active part of it. This reduces the amount of energy wasted on idle listening, in which nodes wait for potentially incoming messages, while still maintaining a reasonable throughput. They discuss the design of T-MAC, and provide a head-to-head comparison with classic CSMA (no duty cycle) and S-MAC (fixed duty cycle) through extensive simulations. Under homogeneous load, T-MAC and S-MAC achieve similar reductions in energy consumption (up to 98 %) compared to CSMA. In a sample scenario with variable load, however, T-MAC outperforms S-MAC by a factor of 5. Preliminary energy-consumption measurements provide insight into the internal workings of the T-MAC protocol.

The S-MAC [5], a MAC protocol designed for wireless sensor networks. Wireless sensor networks use battery-operated computing and sensing devices. A network of these devices will collaborate for a common application such as environmental monitoring. These characteristics of sensor networks and applications motivate a MAC that is different from traditional wireless MAC such as IEEE 802.11 in several ways: energy conservation and self-configuration are primary goals, while per-node fairness and latency are less important. S-MAC uses a few novel techniques to reduce energy consumption and support self-configuration. It enables low-duty-cycle operation in a multi-hop network. Nodes form virtual clusters based on common sleep schedules to reduce control overhead and enable traffic-adaptive wake-up. S-MAC uses in-channel signaling to avoid overhearing unnecessary traffic. Finally, S-MAC applies message passing to reduce contention latency for applications that require in-network data processing. The measurement results of S-MAC performance on a sample sensor node, the UC Berkeley Mote, and reveals fundamental tradeoffs on energy, latency and throughput. The duty cycle is fixed in such protocols, the network throughput can degrade under heavy traffic, while under light loads, unwanted energy consumption can occur. In the proposed Pattern-MAC (PMAC [6]) protocol, instead of having fixed sleep wakeups, the sleep-wakeup schedules of the sensor nodes are adaptively determined. The schedules are decided based on a node's own traffic and that of its neighbors.

II. EXISTING MAC PROTOCOL FOR WSN

A. S-MAC

1) *Basic Characteristics:* A wireless sensor network MAC protocol must have the characteristics described below in order to be suitable for applications on these sorts of networks:

a) Energy efficient in such a way that nodes spend the least amount possible of energy in their communication and computation functions. Communication procedures consume much more energy than those of computation, so this protocol concentrated on communication issues.
b) Scalable to allow the network which is running it to grow without compromising too much its performance.
c) Adaptable to changes not only due to new nodes that enter the network but also due to nodes that fail in their normal operation.
d) Major sources of energy wastage are identified and described below:
e) Collisions: packets discarded have to be retransmitted, which increase energy consumption.
f) Overhearing: a node listens to packets destined to other nodes
g) Control packet overhead: transmission and reception of control packets consumes Energy.
h) Idle listening: Listening to receive possible packets that are not sent.
i) Over emitting: Transmission of a packet when destination node is not ready.

S-MAC takes into account characteristics described above taking more emphasis on reducing energy consumption by reducing idle listening and control packet overhead. This protocol trades off energy consumption with latency. Low latency is a desirable feature for traditional networks that do not have the constraint of energy.

2) *Synchronization:* To minimize the idle listening energy consumption, S-MAC uses a sleep-listen schedule where a node remains inactive for a long time and then wakes up to transmit or receive packets. When slept, a node turns off its radio saving a lot of energy. Figure 2.1 shows S-MAC sleep-listen schedule. A frame is a complete cycle of listen and sleep times. The duty cycle is defined as the ratio of the listen interval to the frame length.

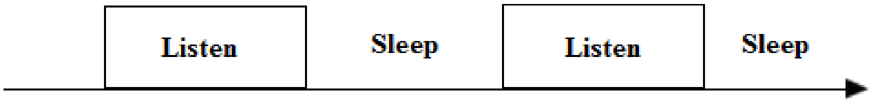

Fig.2.1. Basic Sleep-Listen schedule

Nodes synchronize each other by periodically broadcasting Synchronization (SYNC) packets. A node could follow more than one schedule if received more than one SYNC packet when synchronizing for the first time. In such case, the node follows both schedules. SYNC packets are very short and include the address of the sender and the time of its next sleep.The time of the next sleep is relative to the moment that the sender starts to send its SYNC packet rather than absolute. SYNC packets and data ones are sent on different time slots as shown in Figure 2.2.

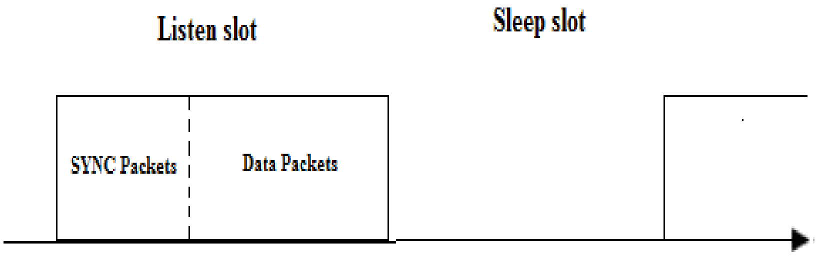

Fig.2.2. The Listen slot is divided

3) *Collision Avoidance:* To avoid collisions, S-MAC uses the 802.11 mechanism: RTS/CTS/DATA/ACK. Only broadcast packets do not follow RTS/CTS/DATA/ACK sequence. A second mechanism that S-MAC uses to avoid collision is the vector called Network Allocation Vector (NAV). In such vector, it keeps the duration of transmission of node's neighbors. In this way, the node knows when the medium will be most likely idle to transmit a data packet. This mechanism is called virtual carrier sense. Furthermore, S-MAC uses Carrier Sense (CS) before transmitting a SYNC or data packet to verify if whether there are current transmissions or not, this mechanism is a physical sense. Within a frame, a node can transmit only a SYNC packet, only a data packet or both. This is shown in Figure 2.3. The figure highlights the procedure RTS/CTS/DATA that S-MAC uses to send a data packet by avoiding collisions.

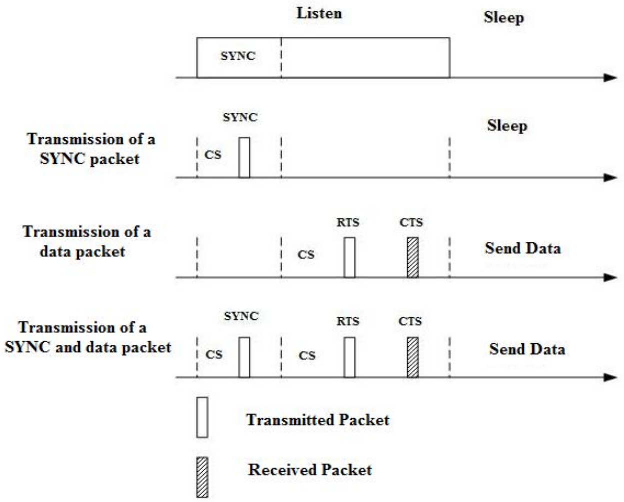

Fig.2.3 Ways to send a data packet and synchronize other nodes.

4) *Adaptive Listening*

Adaptive listening is a mechanism to turn nodes from low-duty cycle to a more active state to reduce latency trading it off with more energy consumption. The basic idea is to let the node who overhears its neighbor’s transmissions (ideally only RTS or CTS) wake up for a short period of time at the end of the transmission. In this way, if the node is the next-hop node, its neighbor is able to immediately pass the data to it instead of waiting for its scheduled listen time. If the node does not receive anything during the adaptive listening, it will go back to sleep until its next scheduled listen time. If the next-hop node is a neighbor of the sender, it will receive the RTS packet. If it is only a neighbor of the receiver, it will receive the CTS packet from the receiver. Thus, both the neighbors of the sender and receiver will learn about how long the transmission is from the duration field in the Request-to-Send (RTS) and Clear-to-Send (CTS) packets. In this way, they are able to adaptively wake up when the transmission is over.

It should be noted that not all next-hop nodes can overhear a packet from the previous transmission, especially when the previous transmission starts adaptively, i.e., not at the scheduled listen time. So if a sender starts a transmission by sending out an RTS packet during the adaptive listening, it might not get a CTS reply. In this case, it just goes back to sleep and will try again at the next normal listen time.

5) *Overhearing Avoidance*

Inspired by PAMAS [10], S-MAC tries to avoid overhearing by letting interfering nodes go to sleep after they hear an RTS or CTS packet. Since DATA packets are normally much longer than control packets, the approach prevents neighboring nodes from overhearing long DATA packets and following ACKs. All immediate neighbors of both the sender (node C) and receiver (node D) should sleep after they hear the RTS or CTS until the current transmission is over, as indicated by “X ” in Figure 2.4.

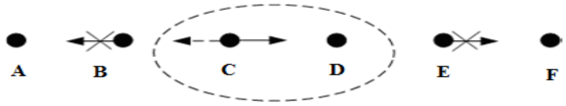

Fig. 2.4. Nodes that should go to sleep when C transmits to D

6) *Message Passing:* Message Passing consists in fragmenting the long message into many small fragments, and transmitting them in a burst. Only one RTS and one CTS are used. They reserve the medium for transmitting all the fragments. Every time a data fragment is transmitted, the sender waits for an ACK from the receiver. If it fails to receive the ACK, it will extend the reserved transmission time for one more fragment, and re-transmit the current fragment immediately. If a neighboring node hears an RTS or CTS packet, it will go to sleep for the time that is needed to transmit all the fragments. Each data fragment or ACK also has the duration field. In this way, if a node wakes up or a new node joins in the middle of a transmission, it can properly go to sleep no matter if it is the neighbor of the sender or the receiver. If the sender extends the transmission time due to fragment losses or errors, the sleeping neighbors will not be aware of the extension immediately. However, they will learn it from the extended fragments or ACKs when they wake up.

B. T-MAC

Timeout-MAC (T-MAC) is proposed to enhance the performance results of S-MAC protocol under variable traffic load. In T-MAC, listen period ends when no activation event has occurred for a time threshold (TA). This operation makes T-MAC's schedule variable instead of the fixed schedule proposed in S-MAC. Figure 2.5 depicts the basic operation of TMAC.

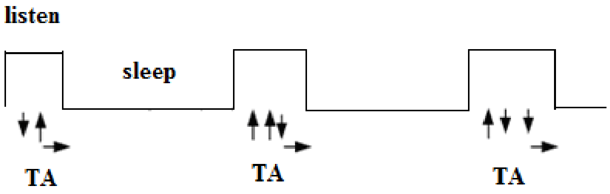

Fig.2.5. T-MAC basic operation

T-MAC synchronization is similar to that of S-MAC. SYNC packets are exchanged between nodes to form virtual clusters that share the same synchronization. A node can run more than one synchronization scheme. The scheme used to contend for the medium is the well known RTS/CTS/DATA/ACK. However, T-MAC proposes a change in this scheme that is used to avoid the early sleep problem. The early sleep problem is the excessive contention for a node that wants to transmit to its neighbors. To avoid this problem T-MAC proposes two solutions: Future request to send and priority on full buffers.

C. D-MAC

The main objective of DMAC is to achieve very low latency and still be energy efficient. For that purpose it is designed to overcome S-MAC problems: increased delivery latency because a packet cannot reach the sink in a single listen period, fixed duty cycles that do not adapt to traffic changes and the increased possibility of collisions due to synchronous duty cycle. It identifies a problem that exists in implicit sleep delay reducing protocols: SMAC and T-MAC. Sleep delay refers to the time that a packet suffers since it is transmitted from originating node to the sink (Base Station). This delay refers to the time that transmitting node has to wait for intended receiving node to wake up and receive the packet. It is shown that S-MAC and T-MAC only solve this problem for a two-hop path. If a network is a multi-hop one with more than 2 hops, the solutions provided by S-MAC and TMAC are not appropriate. This problem is identified as Data forwarding Interruption (DFI).It identified that the limited overhearing range due to radio sensitivity is the origin of this DFI problem.

The solution proposed is a staggered active/sleep schedule shown in Figure 2.6. Low latency is achieved by assigning subsequent slots to nodes that are successive in the data transmission path. With this scheme it is expected that a packets do not suffer from sleep delay at all because the next intended receiving node must always be awake when transmitting node wants to transmit a packet to it.

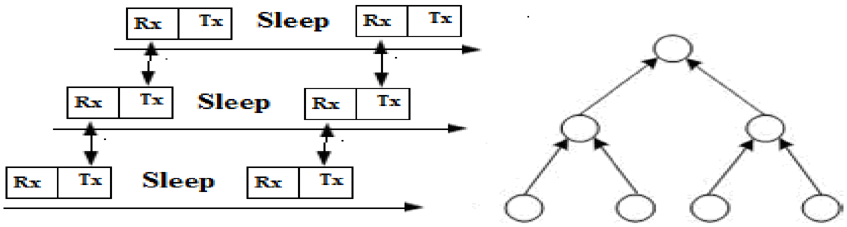

Fig. 2.6. Staggered active/sleep schedule in D-MAC

DMAC staggers the active/sleep schedule of the nodes in the data gathering tree according to its depth in the tree. This allows continuous packet forwarding flow in which all nodes on the multi-hop path can be notified of the data delivery in progress as well as any duty-cycle adjustments.

D. B-MAC

B-MAC is designed for an Ad-Hoc network of nodes with N-sender to 1-receiver transmissions. The basic idea of B-MAC is to keep the protocol simple. That allows very small implementations, an important point because of the limited available memory. Like the other protocols B-MAC uses periodically sleep/wakeup cycles. The mechanism used here is called Low Power Listening. LPL means in the wakeup time the node listens for incoming data transmissions. If there is no data received, called a "false positive", a timeout interrupts the listen state. Otherwise the node waits for complete packet transmission. To ensure that the received packet is complete from the beginning there is a preamble time of 100ms added after the wakeup. Fairness is not guaranteed by LPL. The sleep periods of the nodes can differ to each other, B-MAC is asynchronous. When there is data to send a node switches the radio mode and starts to send an announcement. This announcement must be long enough to make sure that the receiver notices, even if the receiver starts sleeping at the beginning. Afterwards the sender transmits the target address and starts sending data. Asynchronous networks don't need complicated and expensive synchronization methods. There is no data fragmentation used in B-MAC. This would be more complicated to coordinate and B-MAC expects short messages like the ones commonly used for sensor information's. Another concept to reduce the amount of needed energy is clear channel assessment (CCA). This is used for clear channel detection. For energy reduction a better separation between signals and noise on the channel is useful. Therefore the noise must be analyzed. In case of a false positive a sample was put into a queue. It makes sense to capture and analyses more than one sample because the noise caused by the environment changes continuously. An optional feature is using acknowledgements. B-MAC has an application interface for flexible configuring parameters like this. Other options are for example the check interval. A good value for this sometimes depends on the use case so this can be adjusted by a higher layer application.

E. Wise MAC

Wise MAC [9] is an infrastructure protocol. It assumes there is one central unit with unlimited energy supply and connection to another high speed network e. g. Ethernet to exchange data packets. That is why this unit is called access point. Because of the independence to batteries the access point should manage the network. Therefore it has a table containing the wakeup times for every known node. That reduces the needed listening time. These times are part of acknowledgement packets send from the nodes. If the access point has information for a specific node it can be notified as soon as possible. There must be a little delay before message passing because of the possible clock drift to the nodes. The messages are expected to be short for sensor nodes.

The base station handles one message after the other. Synchronized nodes would have to wait except for one node. That is why the wakeup schedules of the nodes are asynchronous. The packet header contains a frame pending bit. It indicates there is more data waiting. After sending an acknowledgement the node stays awake if the bit is set. The sender sends the second packet ongoing to the received ACK-packet.

F. Problem Statement

Existing solutions for the energy-hole problem are dependent on proper node distribution strategies. However, these protocols function in vain in environments where planned node deployment is not possible. In such cases, since network lifetime extension is still necessary, other power-saving protocols (such as S-MAC, T-MAC, and DMAC [7]) should be applied. One of the problems of sensor nodes running S-MAC, T-MAC, or DMAC is that they have to wake up at every time frame to check if there is pending traffic. Since sensor nodes have different traffic loads (nodes that are closer to the sink are the heavy-loaded ones), all nodes adopting the same frequency to listen/sleep are not power efficient. PMAC allows nodes to have different listen/sleep frequencies, but its functionality relies on the correct exchanges of the nodes' schedules. A scheme that lets sensor nodes choose their own wake-up frequencies is desirable, but the challenge is that sensor nodes may not be able to meet each other if they are not well controlled. For example, as shown in Fig.2.7, two nodes, i.e., A and B, with the same wake-up frequency of 0.5 (wake up every two time frames) will never meet if host B always wakes up one time frame later than host A. In other words, what we need is a power efficient protocol that can alleviate the energy-hole problem when intentional node deployment is difficult or not impossible. The solution should not only be able to adjust each sensor node's listen/sleep frequency according to their traffic loads, but it should also guarantee that sensor nodes meet each other.

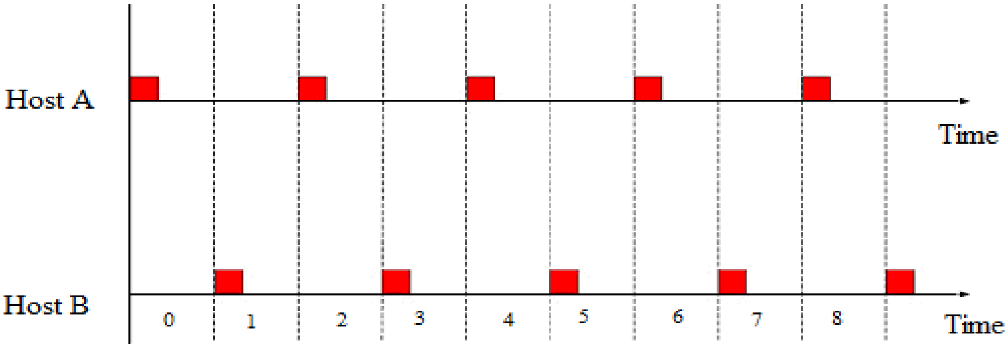

Fig.2.7. Example that two hosts that never meets.

III. QUORUM-BASED ENERGY SAVING MAC PROTOCOL

A. *Quorum Concept*

A quorum is a request set that enables some actions if permission is granted. There are always nonempty intersections between any two quorum sets. There are many kinds of quorum, such as the majority-based quorum, the tree-based quorum, the grid-based quorum, and others. Here a quorum set represents the time frames wherein a sensor node must wake up. For non-quorum time frames, sensor nodes are allowed to enter sleep mode for the entire time frame to conserve energy. Because quorums are used, any two nodes can wake up and meet each other at some time frame. In a grid-based quorum, one row and one column are picked in an $n \times n$ grid as a quorum set. This concept is shown in Figure 3.1. Host *A* picks row R_a and column C_a as its quorum, while host *B* picks row R_b and column C_b. There are two intersections between hosts *A* and *B*: one for R_a and C_b and the other for C_a and R_b. As mentioned earlier, sensor nodes must wake up at their chosen quorums. This means that both nodes will wake up at these intersections.

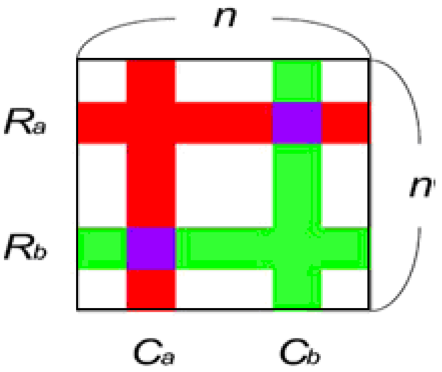

Fig. 3.1 Example of a grid-based quorum.

Fig. 3.2 is an example of quorum time frame selection, where node *A* picks the third row and the third column as its quorum set, while host *B* selects the first row and the first column. That is, node *A* wakes up at time frames 2, 5, 6, 7, and 8, while node *B* wakes up at time frames 0, 1, 2, 3, and 6.The intersections are frames 2 and 6 when both nodes *A* and *B* are awake.

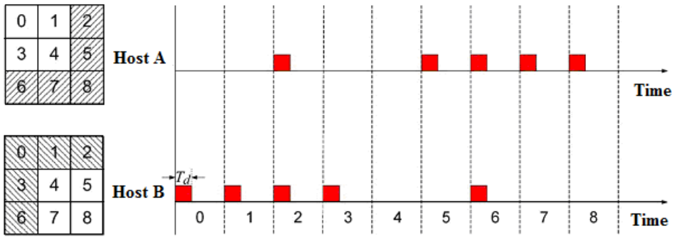

Fig.3.2. Example of intersections. Host *A* & host *B* meet each other at intervals 2 and 6.

B. *Quorum-Based Wake-Up Schedule*

The QMAC protocol aims to reduce power consumption and determine the sleep frequency for each sensor node based on its own traffic load. Each sensor node randomly and independently selects one row and one column as its quorum set. In QMAC, power saving is achieved by reducing the number of wakeup time frames. As mentioned earlier, we use a quorum set to represent the time frames wherein a sensor node must wake up. However, during these wake-up time frames, a sensor node does not always stay awake for the entire time frame. A sensor node can go to sleep whenever it identifies that another transmission that it is not involved in is activated. Furthermore, a sensor node will go to sleep if the channel is idle for duration of *Td*. In this paper, the value of *Td* is set to one fifth of the length of a time frame. A sensor node using an $n \times n$ grid will wake up $2n – 1$ out of $n2$ time frames. Because they have heavier traffic, sensor nodes located in the inner coronas can use a smaller grid. On the other hand, sensor nodes in outer coronas can pick up a larger grid. By choosing grid sizes based on each sensor node’s individual traffic load, QMAC solves the fixed-listen/sleep frequency problem. Note that sensor nodes that use different quorum sizes are still guaranteed to meet each other. Thus, it is feasible for QMAC to operate in WSNs. Furthermore, note that a sensor node with pending traffic will not go to sleep until the traffic is delivered, regardless of whether a quorum time frame is coming or not.

Next, we describe how we decide on the exact grid size for each sensor node. Since we assume a many-to-one and constantly reporting model, it is possible to determine each sensor node's grid size in advance. A sensor node that is in an inner corona has more traffic because, aside from its own traffic, it has to relay traffic from nodes in outer coronas. To obtain the exact traffic load, we need to know the average number of outer nodes of which an inner node is in charge. Here we describe our QMAC using a four-corona network, as shown in below Fig.3.3.

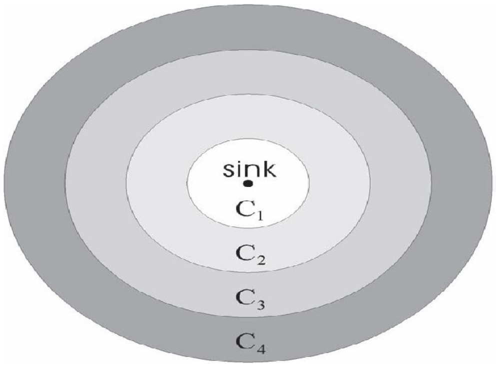

Fig.3.3 Network divided into adjacent coronas centered at the sink node.

The traffic load of the node is depends on the area of the coronas. The ratio of areas for different coronas $C1 : C2 : C3 : C4$ is 1 : 3 : 5 : 7. This means that, on the average, a node in $C3$ is responsible for relaying traffic for 7/5 nodes in $C4$, a node in $C2$ is responsible for 5/3 nodes in $C3$, and a node in $C1$ will take care of three nodes in $C2$.

Assume that the average hop distance is R. The area of $C1$ is πR^2, and the area of $C2$ is $\pi(2R)^2 - \pi R^2 = 3\pi R^2$. Similarly, the areas of $C3$ and $C4$ are $5\pi R^2$ and $7\pi R^2$, respectively. The area of the corona = $\pi(\text{corona number} * R)^2$ – area of previous corona

Assuming that each node generates one unit of traffic for each reporting, the sensor nodes in $C4$ only have their own traffic to deliver; thus, their traffic load is one. Aside from their own traffic, sensor nodes in $C3$ have to forward traffic from $C4$. Each node in $C3$ is responsible for 7/5 nodes in $C4$; thus, the traffic load in $C3$ is $1 + (7/5) \times 1 = 2.4$. Similarly, nodes in $C2$ and $C1$ have traffic loads of $1 + (5/3) \times 2.4 = 5$ and $1 + 3 \times 5 = 16$, respectively. In general, in a network where sensor nodes generate the same amount of traffic, the traffic load for sensor nodes in Ci can be denoted by TCi and calculated by

$$T_{ci} = \{1 + [C_{i+1}] / [C_i]\} * T_{ci+1\backslash}$$

Where $\left|C_i\right|$ means the area of Ci. In a network with different numbers of coronas, the grid sizes can be calculated in the same way.

It is reasonable to assign a grid size based on a sensor node's traffic load. If sensor nodes in $C1$ use a 2×2 grid, then their ratio of wake-up time frames is 0.75. According to the ratio of traffic loads for different coronas ($C1 : C2 : C3 : C4 = 16 : 5 : 2.4 : 1$), the ratio of wake-up time frames for the sensor nodes in $C2$, $C3$, and $C4$ should be 0.234, 0.112, and 0.047, respectively. This indicates that the quorum sizes used by $C2$, $C3$, and $C4$ should be 8×8, 17×17, and 42×42, respectively.

C. Latency Reduction

In allowing sensor nodes to sleep longer than one time frame, QMAC is believed to reduce energy consumption. The price for this saved energy, though, is higher latency. Assuming that sensor node A in corona Ci relies on the help of sensor node B in corona $Ci-1$ to relay its data. Longer latency is induced since the quorum of node B may not coincide with that of node A. Thus, node A may take several time frames before it meets node B. To reduce the latency induced when applying QMAC, we introduce the concept of "next-hop group." Each sensor node, e.g., node X, in Ci randomly selects a set of sensor nodes that are both within $Ci-1$ and in the coverage range of node X as its candidate relaying nodes, as shown in Figure 3.4. This set of candidates form the next-hop group of node X. The next-hop group selection for each sensor node can be done in the network-initialization phase, where each node notifies its next-hop group members through a control packet MEMBER_NOTIFY. The basic idea of the next-hop group is to make more sensor nodes in inner coronas, instead of only one, capable of relaying packets for a particular sensor node in outer coronas. Thus, when a sensor node in an outer corona reports its data, it already has multiple candidate next-hop nodes and has a good chance of meeting one faster. It should be noted that only one member in the next-hop group is selected to relay traffic for each data report. A node can reselect its next hop group member by retransmitting a MEMBER_NOTIFY packet whenever the node finds that its next-hop group does not function properly.

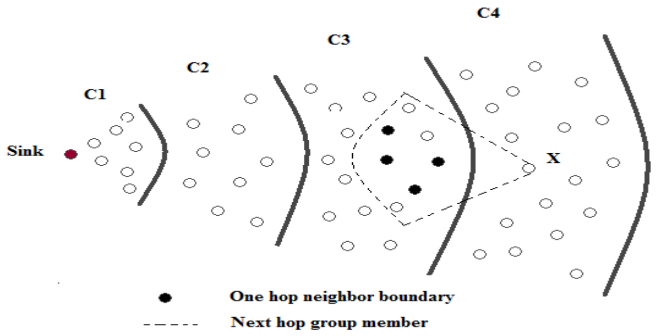

Fig.3.4. Next-hop group

The data reporting of QMAC_LR follows the four-way “request-to-send (RTS)/CTS/DATA/ACK” dialog. When sensor node X reports its data, the relay-node-selection process occurs as follows.

1) Sensor node X multicasts an RTS to its next-hop group members. One possible way to implement multicasting is to set the multicast address as the address of the source node. This means that an RTS packet with the same source and destination address will indicate a multicast transmission.
2) All the awake members of X that receive the RTS will back off before replying with a CTS packet. The back-off time selection is based on each node’s remaining battery capacity. The ones that have more than half of their remaining battery capacity will randomly distribute their back-off time at the first half of the contention window. On the other hand, the ones that have less than half of their remaining battery capacity will locate their back-off time at the second half of the contention window.
3) After node X receives a CTS from one of its next-hop group members, e.g., node Y, it can transmit its data to Y (through unicast). This data transmission also informs the other awake members that the relay node has been selected, and they can cease their back-off processes and go to sleep. Finally, correctly receiving data from node X, node Y will reply with an ACK packet.

We use an example to illustrate the operation of data transmission with latency reduction. As shown in Fig.3.5, sensor node A initiates the dialog with a multicast RTS packet. Sensor nodes B, C, and D, being the next-hop group members of node A, will accept the RTS packet, while sensor node E, which is not a next-hop group member of node A, will simply ignore it and go to sleep. After contention resolution, node C then transmits a CTS packet back to A. Once the DATA packet from node A is received, all nodes other than node C will enter sleep mode.

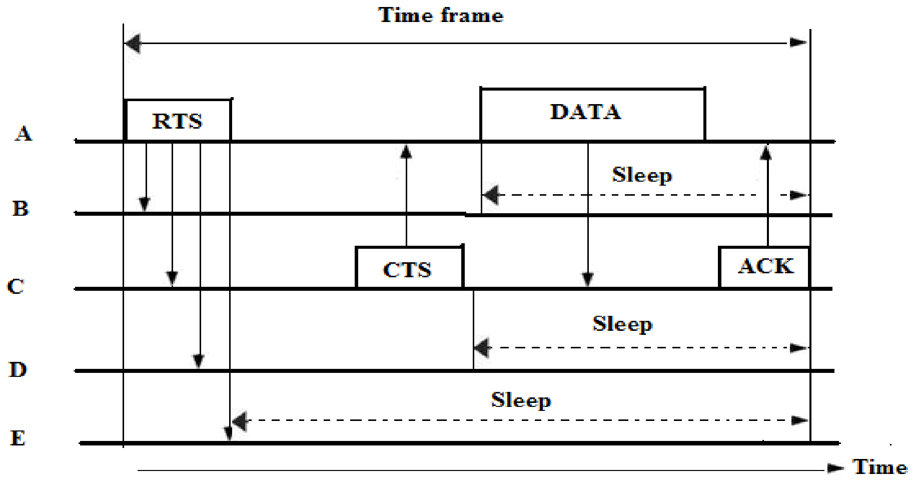

Fig3.5 Example of the operation of latency reduction.

The proper setting of next-hop group sizes is critical to the success of latency reduction. To reduce power consumption, the size should be as small as possible. The drawback of a small next-hop group is longer latency, while a large next-hop group may produce many redundant candidate sensor nodes, which induces energy waste. To find the proper size of a next-hop group, we need to know the probability wherein a sensor node, with the intention to report, meets at least one of its next-hop members.

IV. SIMULATION & DESIGN

A. The Network Simulator (Ns2)

NS is an event driven network simulator developed at University of California at Berkeley, USA, as a REAL network simulator projects in 1989 and was developed at with cooperation of several organizations. Now, it is a VINT project supported by DARPA. NS is not a finished tool that can manage all kinds of network model. It is actually still an on-going effort of research and development. The users are responsible to verify that their network model simulation does not contain any bugs and the community should share their discovery with all. There is a manual called NS manual for user guidance.

NS is a discrete event network simulator where the timing of events is maintained by a scheduler and able to simulate various types of network such as LAN and WPAN according to the programming scripts written by the user. Besides that, it also implements variety of applications, protocols such as TCP and UDP, network elements such as signal strength, traffic models such as FTP and CBR, router queue management mechanisms such as Drop Tail and many more. There are two languages used in NS2. They are C++ and OTcl (an object oriented extension of Tcl). The compiled C++ programming hierarchy makes the simulation efficient and execution times faster. The OTcl script which written by the users the network models with their own specific topology, protocols and all requirements need. The form of output produce by the simulator also can be set using OTcl. The OTcl script is written which creating an event scheduler objects and network component object with network setup helping modules. The simulation results produce after running the scripts can be use either for simulation analysis or as an input to graphical software called Network Animation (NAM).

B. The Network Animation (Nam)

The network animator began in 1990 as a simple tool for animating packet trace data. This trace data is typically derived as output from a network simulator like ns or from real network measurements, e.g., using tcpdump. Steven McCanne wrote the original version as a member of the Network Research Group at the Lawrence Berkeley National Laboratory, and has occasionally improved the design, as he's needed it in his research. Marylou Orayani improved it further and used it for her Master's research over summer 1995 and into spring 1996. The nam development effort was an ongoing collaboration with the VINT project. Currently, it is being developed at ISI by the SAMAN and Conser projects.

C. Parameters Used To Design Qmac Protocol

Simulation of QMAC protocol is performed based on the following parameters.
Nodes = 15
Maximum transmission range = 50m
Circular area = 200m
Channel capacity = 10kbps
CBR payload = 128B @ 2.5s with maximum delay of 100ms
Power mode threshold:
Transmit = 0.66w
Receive = 0.395w
Sleep =0.35w
Simulation time= 20/40/100 ms

D. Simulation Results

1) Nam For Qmac Protocol

Figure 4.1 NAM for QMAC protocol

2) Impact On Live Nodes

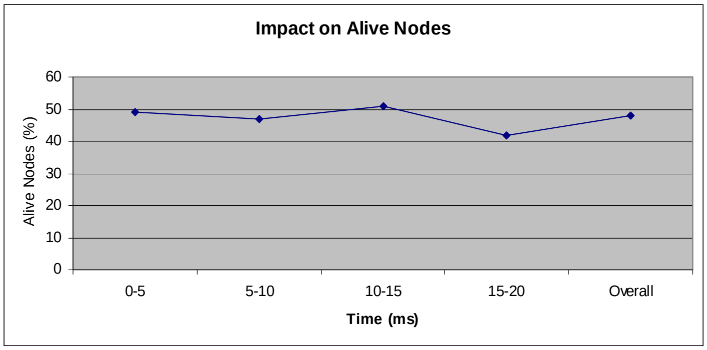

Fig. 4.2 Impact on Live Nodes

The Fig. 4.2 shows the performance of the proposed protocols in terms of how much they can extend network lifetime. The proposed protocol will increase the life time of the node by increasing the sleep duration more than one time interval.

3) Impact On Transmission Latancy

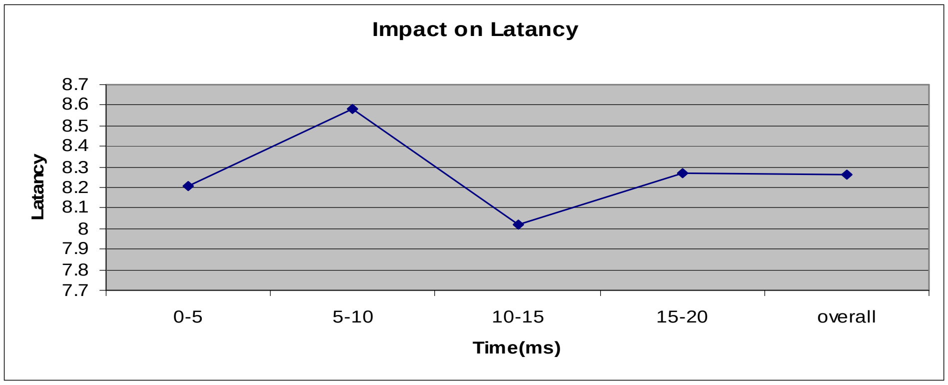

Fig. 4.3 Effect of QMAC protocols on latency.

Transmission latency for QMAC protocols can be found in Fig. 4.3. We separately observed the average latency at durations of 0–5, 5-10, 10-15 and 15-20s, as well as observed the overall average latency.

4) Impact On The Transmission Success Ratio

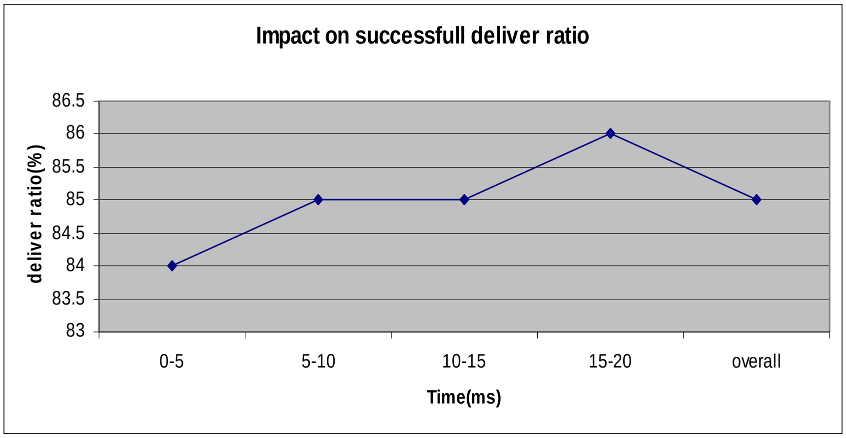

Fig. 4.4. Effect of QMAC protocols on the transmission success ratio.

Transmission success ratio for QMAC protocols can be found in Fig. 4.4. Similar to prior latency experiments, we also separately observed them at 0–5, 5-10, 10-15 and 15-20s, as well as observed the overall average latency.

V. CHAPTER 5CONCLUSION AND FUTURE WORK

Energy conservation is crucial in wireless sensor network. Typically, sensor nodes closer to the sink run out of energy faster. Most previous solutions have tried to deploy more nodes around the sink. Here we have focused on an environment where planned deployment is difficult, and we have also proposed an energy-conserving MAC protocol. Realizing that sensor nodes have different loads due to their different distances to the sink, we have applied the concept of quorum to enable sensor nodes to adjust their sleep durations based on their traffic loads. To reduce delays induced by longer sleep durations, we have increased each node's transmission opportunity by enabling a group of next-hop nodes to accomplish the packet-relaying job. Such a mechanism also enhances the robustness of the proposed protocol. Simulation results verify that our MAC reduces energy consumption and keeps the latency low. This reveals that MAC is a promising energy-saving protocol for randomly deployed sensor networks. In future i am going to design this MAC protocol with mobility nodes.

REFERENCES

[1] C.-M. Chao, J.-P.Sheu, and I.-C. Chou, "An adaptive quorum-based energy conserving protocol for IEEE 802.11 ad hoc networks," IEEE Trans. Mobile Comput., vol. 5, no. 5, pp. 560–570, May 2006.
[2] J.-R. Jiang, Y.-C.Tseng, C.-S. Hsu and T.-H. Lai, "Quorum-based asynchronous power-saving protocols for IEEE 802.11 ad hoc networks," Mob. Netw. Appl. (MONET), vol. 10, no. 1/2, pp. 169–181, Feb. 2005.
[3] J. Li and P. Mohapatra, "An analytical model for the energy hole problem in many-to-one sensor networks," in Proc. IEEE 62nd Veh. Technol. Conf., Dallas, TX, Sep. 2005, pp.
[4] T. V. Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in Proc.ACM SenSys, Los Angeles, CA, Nov. 2003,
[5] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in Proc. 21st Int. Annu. Joint Conf. IEEE Comput. Commun. Soc. INFOCOM, New York, Jun. 2002, pp. 214–226.
[6] T. Zheng, S. Radhakrishnan, and V. Sarangan, "PMAC: An adaptive energy-efficient MAC protocol for wireless sensor networks," in Proc. IEEE Int. Parallel Distrib. Process. Symp., Denver, CO, Apr. 2005, pp. 65–72.
[7] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy efficient and low-latency MAC for tree-based data gathering in sensor networks," in Proc. 18th Int. Parallel Distrib. Process. Symp, Santa Fe, NM, Apr. 2004, pp. 224–231.
[8] M. Buettner, G. Yee, E. Anderson, and R. Han, "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks," in Proc. 4th Int. Conf. Embedded Netw. SenSys, Boulder, CO, Oct. 2006, pp. 307–320.
[9] A. El-Hoiydi and J.-D. Decotignie, "Low power downlink MAC protocols for infrastructure wireless sensor networks," Mob. Netw. Appl., vol.10, no.5, Oct. 2005.
[10] S. Singh and C.S. Raghavendra, "PAMAS: Power aware multi-access protocol with signaling for ad hoc networks," ACM Computer Communication Review, vol. 28, no. 3, pp. 5–26, July 1998.