Chapter 3: Medium Access Control in Wireless Sensor Networks

3.1 Introduction

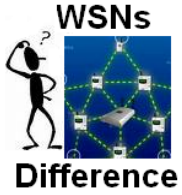
A wireless sensor network (WSN) is a collection of different sensor nodes used to sense the environment for applications such as vibration, temperature, pressure, sound, and pollutants in the environment. In the WSNs, each sensor node is an autonomous device which consists of communicating device, computing device, sensing device and memory. To effectively exchange data among multiple sensor nodes, WSNs have to employ the medium access control (MAC) protocol to coordinate the signal transmissions over the shared wireless radio channel. Otherwise, multiple nodes may try to access the transmission medium (e.g., wireless channel) simultaneously, which leads to signal collision, data loss, retransmission, wastage of energy, delay in the data transmission, and so on.

Remember

A medium access control (MAC) protocol determines how multiple nodes share access to a physical medium (e.g., wireless channel), by defining the communication schedules and rules such as (1) which nodes should occupy the channel; (2) when and how long the nodes can occupy the channel; (3) how the nodes use the channel to talk with its neighbor nodes.

Why do WSNs require Medium Access Control?

MAC protocols play a vital role in any network paradigms including wired networks, Mobile Ad hoc Networking (MANET), and wireless sensor networks. However, the design of efficient MAC protocols has to take into account the unique challenges emerged in respective networking paradigm. For example, unlike the wired networking such as Ethernet, a wireless channel in WSNs generally experiences more data loss due to collision, signal loss, noises and even link breakage. The signal collision occurs in the wireless link cannot be detected the same way as that in the wired link. In addition, a WSN owns very limited resources such as energy, bandwidth and computing capability, which constraints the applicability of the MAC protocols developed in other wireless networks including Wi-Fi and MANET.



In general, the medium access control schemes developed in other network paradigms cannot be directly applied in wireless sensor networks due to the unique and challenging issues posed by the wireless medium, the tiny sensor nodes, and various WSN applications.

MAC Design is challenging in WSNs

As a specific type of wireless networks, the WSN shares similar challenges faced in other wireless networking technologies. As elaborated below, such challenges and many other resource constraints in WSNs have significant impacts on how medium access control is conducted in sensor nodes [Awoo01].

1. Resource constraints

2. Signal loss in wireless channel
3. Collision at the receiver's end
4. Hidden and exposed terminal problems

1. Resource Constraints

As described in previous chapter, a wireless sensor node owns limited resources such as power, bandwidth, computing capability and storage space, which must be taken into account when devising MAC protocols in WSNs. Energy is a key concern in battery-powered sensor node. Once the battery is consumed, it is generally difficult or impractical to charge/replace exhausted batteries. That is why, the primary objective in many WSN MAC protocol design is maximizing node/network lifetime, leaving the other performance metrics as secondary objectives. For example, the energy could be saved by turning off the devices which are not in use at the particular period of time.

Good Idea

Since the communication of sensor nodes is much more energy consuming than the computation, minimizing the communication while achieving the desired network operation has been one popular and effective approach to design energy-efficient WSN MAC protocols.

The bandwidth in a WSN is pretty low when compared to that of wired networks such as fiber optical networking. The bandwidth constraint and the dynamics of WSN topology also impose challenging issues to be considered in MAC design. Specifically, in WSNs, the data is sensed and stored in distributed fashion and every sensor node is an autonomous device which is

independent of other nodes in the network. Sensor nodes need to communicate with one another to self-organize as a network system for data transmission whereas redundancies should be avoided. Moreover, the sensor nodes may fail due to the fact that the tiny sensor nodes are fragile. The topology of the network changes, when the nodes failures occur in the network. Similarly, power depletion and node movement may also result in network topology change.

2. Signal loss in wireless channel

Wireless sensor networks employ wireless channel as the transmission media, which suffer signal distortion and loss due to attenuation, reflection, diffraction, scattering, and so on. The signal attenuation generally refers to the loss of energy as the transmitted signal travels from the source node to the destination node through the air. The transmitted signal can get reflected when there are obstacles between the source node and destination node. The edges of the obstacles can results in multiple signals divided from the original transmitted signal and the rough surfaces of the obstacles can cause scattering due to multiple signal reflections. A commonly used wireless propagation mode with omnidirectional antenna was introduced in [Rappaport96] whereby the signal power received at node *j* from the sender node *i*, is as the following equation.

$$P_j = \beta \frac{P_i}{d_{ij}^{\alpha}} \tag{3.1}$$

In Equation (3.1), P_j, P_i, d_{ij} represents the power received at node *j*, power sent out from node *i*, and the distance between node *i* and node *j*, respectively, while α and β denote the energy loss constant, typically depending on the wireless transmission environment. The above equation also indicates that the longer the wireless signal propagates in the air, the more power loss can

occur. In fact, a wireless node *i* can reach another node *j* (or a wireless link exists from node *i* to node *j*) if and only if node *i* transmits at a certain power level. Otherwise, the receiver *j* cannot properly decode the signal for the transmitted data information from sender *i*, or node *j* cannot hear the signal from node *i* at all due to the power loss. In other words, each sensor node has a limited transmission range. For battery powered WSNs, the transmission range of a node varies dynamically and the wireless links among nodes are susceptible to failures/changes, which necessitates different WSN link access control schemes.

3. Collisions occur at the receiver's end

When two or more sensor nodes send data to other nodes simultaneously through the same channel, multiple signals might collide at the receiver side, which prevents the receiver getting and meaningful data information. To ensure reliable data transmission, MAC protocols have to define processes (e.g., retransmission after random delay) to recover from the collision. Collisions result in wastage of energy, lower bandwidth utilization and larger data deliver latency. In wired networking such as Ethernet, collisions can be easily detected by comparing the sent signal and the received signal at the sender side. Accordingly, the sender (e.g., in Ethernet) concludes that another sender is also sending data and take some operations to recover from the collision quickly. However, in wireless sensor networks, the signal sent from the sender is not equal to that received by the receiver due to signal loss or obstacles. For example, assume that two senders are sending data to the same receiver simultaneously. If the two senders are not within each other's transmission range or there is an obstacle preventing two senders hearing each other, the signals from the two senders collide at the receiver, which cannot be detected by either sender.

4. Hidden terminal and Exposed terminal Problems [Awoo01]

As shown in Figure 3.1, the circle around each node represents the corresponding transmission range of the node when omnidirectional antennas are employed and assume that all nodes have the same transmission range. Two sensor nodes are said to be in mutual range (or in the same collision domain) when the transmission ranges of the two nodes interfere with one another. For example, nodes 1 and 2 are in mutual range. Similarly, nodes 2 and 3 are in mutual range while nodes 4 and 3 share the same collision domain. Obviously, when a node is receiving data from the neighbor, there can be only one valid transmission (or sender) within the node's mutual range. Otherwise, multiple signals will collide at the receiver, which results in data loss and energy wastage. To minimize collisions, carrier sense is widely used in the design of MAC protocols. With carrier sense, the transmitter listens the transmission channel for a carrier signal to detect if there is an ongoing transmission from another node before attempting to send data. If a carrier is sensed or there is an ongoing transmission in the medium, the node can wait for the transmission in progress to finish before initiating its own transmission.

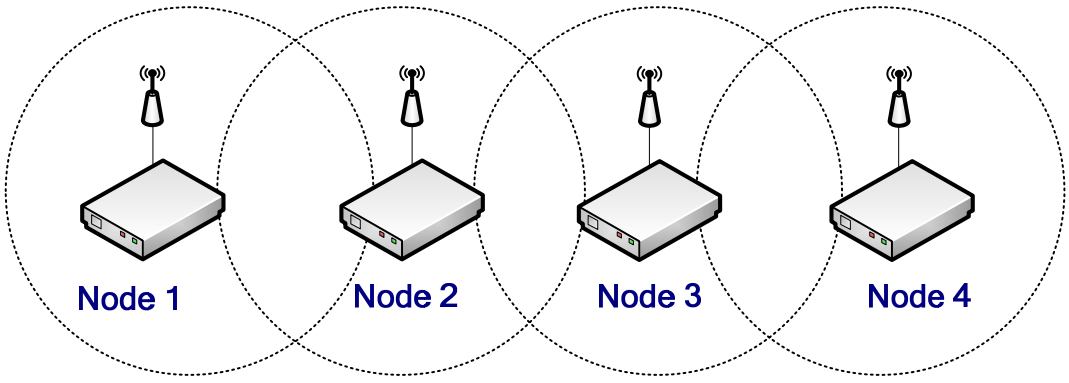


Figure 3.1 Hidden and Exposed Terminal Problems

The carrier sense scheme serves well in Ethernet MAC protocol. However, the significant difference between the signal sent from the sender and the signal received by the receiver makes

carrier sense much ineffective in wireless environment. For example, assume that there is no ongoing communications among the nodes in Figure 3.1. At one time, both sensor node 1 and sensor node 3 sense some events and decide to notify node 2. Before sending data, both node 1 and node 3 sense the channel and learn that the channel is free. Hence, if node 1 and node 3 start to send data to node 2 simultaneously, this leads to data collision at node 2. Even when there is an ongoing transmission from node 3 to node 2, node 1 cannot sense the signal from node 3 due to not sharing mutual range. Subsequently, node 1 may assume the channel is unoccupied and are not aware that node 2 is already engaged in a transmission. The signal from node 1 may disrupt the transmission from node 3 to node 2. This is due to node 3 is invisible to node 1 even though both can reach node 2. This is called as the well-known hidden terminal problem in wireless networking.

The hidden terminal problem indicates that carrier sense in wireless networking may fail in avoiding collisions. In addition, carrier sense can result in channel underutilization in wireless networking.

Consider that node 2 is transmitting data to node 1 and node 3 also intends to send data to node 4. Node 3 performs the carrier sense and finds that the transmission channel is occupied and has to wait the finish of the transmission from node 2 to node 1. However, only the signal interference or collision at the receiver side leads to data loss, energy and bandwidth wastage. In fact, node 3 and node 2 can simultaneously send data to node 4 and node 1, respectively. This is because the interference between the two transmissions (i.e., node 2 to node 1 and node 3 to node

4) does not occur at the receive side. Hence, node 3 is prevented from sending data to node 4 even though both node 4 and node 1 should be able to receive respective data properly. This is called exposed terminal problem in wireless networking.

There has been extensive research in MAC protocol design to resolve above challenges and problems in WSNs (for example, [Ftobagi75, Pkarn90, Bharghavan93]. Several earlier MAC research results on carrier sense and hidden terminal problem in wireless networking is collectively adopted by the IEEE 802.11 standard [IEEE07], which also serves as the starting basis for many MAC protocols proposed for wireless sensor networks. Hence, in the rest of this chapter, we first briefly go through the IEEE 802.11 project. Then we present the classification of medium access control protocols followed by discussions on several typical sensor MAC protocols of each category, which include Sensor Medium Access Control (S-MAC) [Wye02], Timeout Medium Access Control (T-MAC) [Tvdam03], Traffic Adaptive Medium Access (TRAMA) [Vrajendran06], Sift Medium Access Control [Kjamieson03], Zebra MAC (Z-MAC) [Irhee08], and Berkeley MAC (B-MAC) [Jpolasstre04].

3.2 Overview of project IEEE 802.11

Figure 3.2 shows the layered architecture of the IEEE 802.11 project. The physical layer in IEEE 802.11 contains Direct Spread Spectrum (DSSS), Frequency-hopping spread spectrum (FHSS), Infrared, 802.11a, 802.11b and 802.11g. The DSSS defines the physical medium in the frequency of 2.4 GHz ISM band, at the data rates of 1Mbps to 2 Mbps. The FHSS employs the physical media of the same frequency and of the same data rates as that of DSSS. But the basic difference is the number of channels. The number of channels depends on the network regulatory

agencies in every country. In the DSSS, it varies between 13 in European nations and 1 in Japan. While for the FHSS, it varies between 70 in US and 23 in Japan. Similarly, the Infrared has the same data rates as that of FHSS and DSSS. But the Infrared uses the wavelengths in the range of 850 to 950 nm.

The data link layer includes the logical link control and medium access control which defines two access methods: the distributed coordination function (DCF) and the point coordination function (PCF).

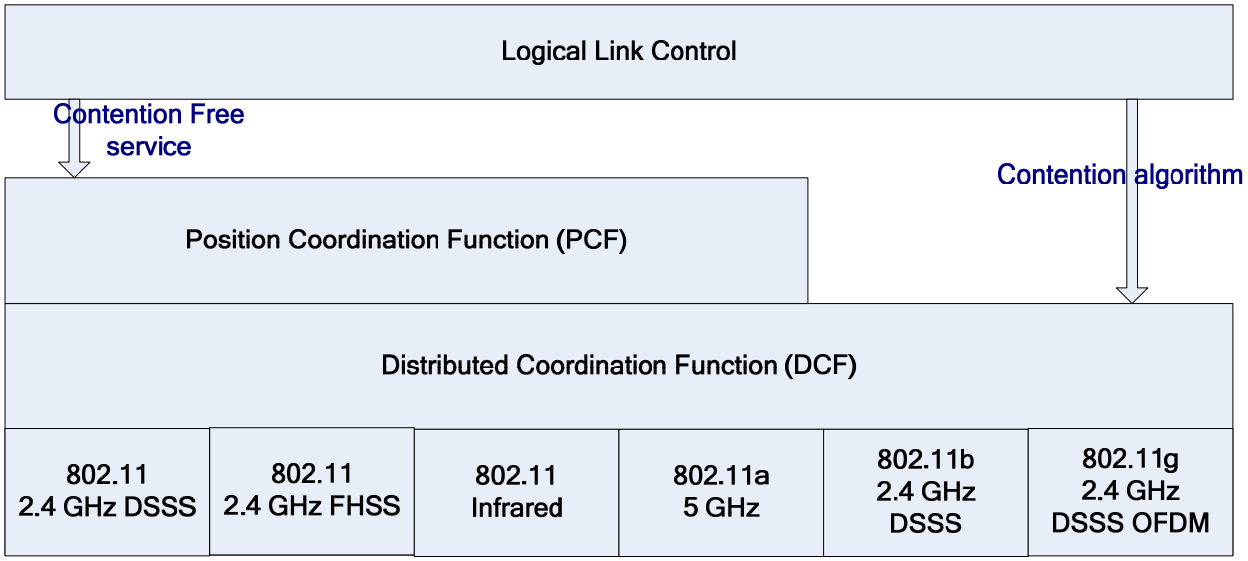


Figure 3.2 IEEE802.11 protocol architecture [Wstallings04]

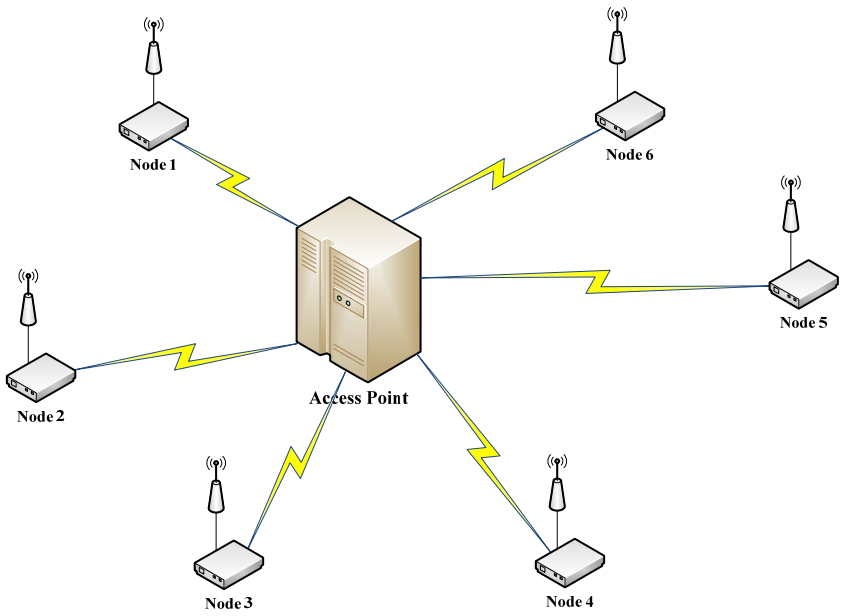


Figure 3.3 IEEE 802.11 infrastructure mode

1. Point Coordination Function (PCF)

The 802.11 MAC defines the point coordination access scheme called Point Coordination Function (PCF) to provide contention-free service, which is available only in the infrastructure mode as shown in Figure 3.3. In the infrastructure mode, stations are connected to the network through an Access Point (AP) which employs a centralized MAC algorithm. This mode can conveniently support high traffic priority.

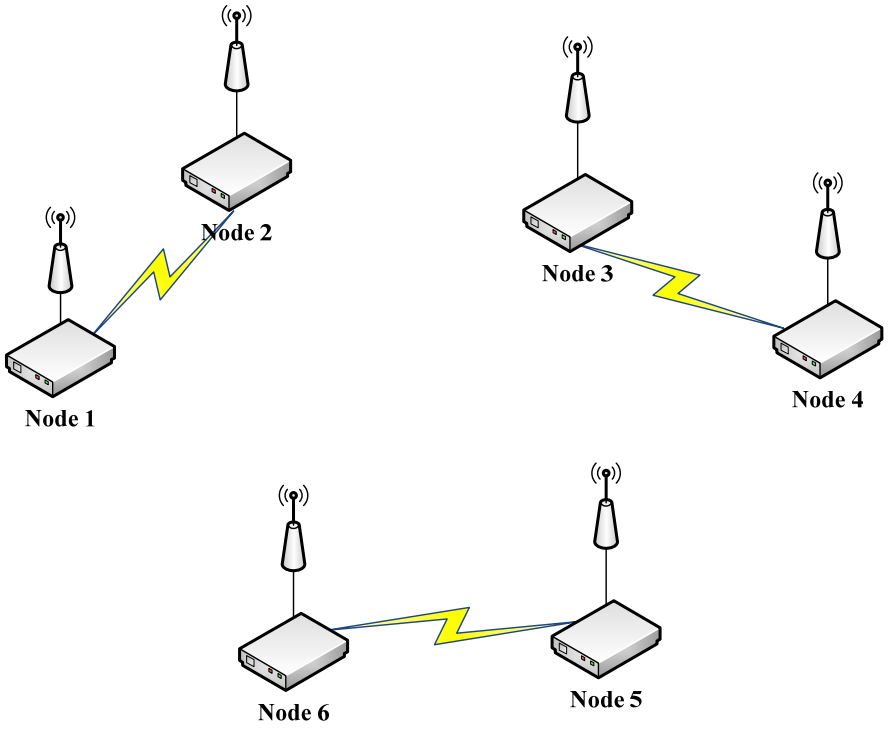


Figure 3.4 IEEE 802.11 ad hoc mode

2. Distributed Coordination Function (DCF)

The Distributed Coordination Function (DCF) is defined to share the medium between multiple stations in an ad hoc mode as shown in Figure 3.4. The DCF enables the stations to exchange data asynchronously by applying Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) and the IEEE 802.11 RTS/CTS to share the medium between stations. CSMA/CA belongs to a class of protocols called multiple access methods [Pkarn90, Bharghavan93]. In CSMA, a station has to first listen to the channel for a predetermined amount of time so as to check whether the channel is free before sending data. If the channel is sensed busy before transmission then the transmission is deferred for a "random" interval to avoid collisions. Note that as mentioned earlier, collision detection is not feasible due to the nature of wireless channel and hidden terminal problem. Hence the scheme of exchanging Request to Send (RTS) packet

and Clear to Send (CTS) packet is introduced in IEEE 802.11 to alert all nodes within range of the sender, the receiver, or both, to keep quiet for the duration of the main data transmission.

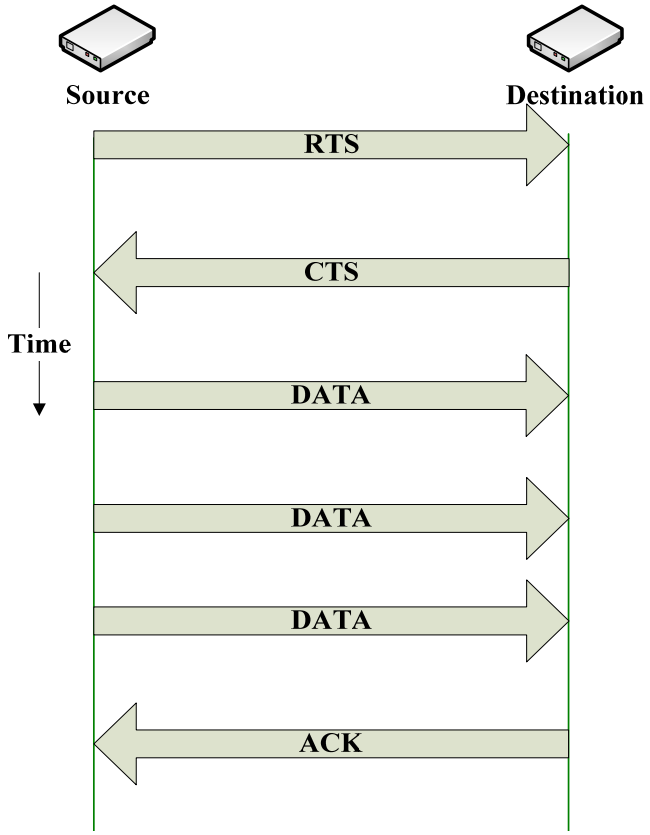


Figure 3.5 Data transfer using RTS-CTS mechanism

As shown in Figure 3.5, the source node (or sender) sends a RTS packet to the destination (or receiver) if the sender wants to send data to the receiver. The destination node replies with a CTS packet. Any other node receiving the RTS or CTS packet should refrain from sending data for a given time to avoid collisions (or solve the hidden node problem). The amount of time the node should wait before trying to get access to the wireless medium is included in both the RTS and the CTS packet. The format of the RTS packet is shown in the Figure 3.6 and the format of CTS packet is shown Figure 3.7.

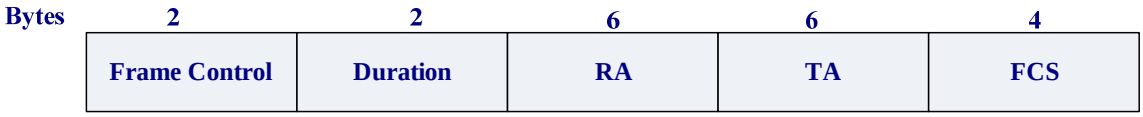


Figure 3.6 RTS Packet Format [IEEE07]

There are five fields in RTS packet format which are:

1. Frame Control (2 bytes): This field contains the information about the version of protocol used, power management, whether there are more fragments of the data and whether the packet is protected or not.
2. Duration (2 bytes): gives the time remaining for transmitting the data or management information plus one CTS frame and one ACK frame.
3. RA (6 bytes): is the address of the intended destination.
4. TA (6 Bytes): is address of the source which initiated the data transfer.
5. FCS (4 Bytes): used for checking for errors in data transmission. It is Cyclic Redundancy Code (CRC) of length 32 bits. It is calculated for all the fields including the header and is calculated using a 32 degree polynomial.

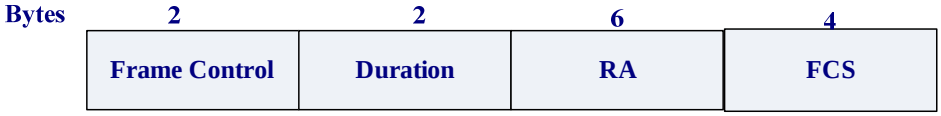


Figure 3.7 CTS Packet Format [IEEE07]

There are four fields in CTS packet format which are:

1. Frame Control (2 bytes): This field contains the information about the version of protocol used, power management, whether there are more fragments of the data and whether the packet is protected or not.
2. Duration (2 bytes): is the difference between the duration field received from the source and the time in CTS frame.

3. RA (6 bytes): is the address of the intended destination which is copied from the TA field in RTS packet format. If the CTS packet is the first packet that the destination is transmitting then RA is the transmitter MAC address.
4. FCS (4 Bytes): used for checking for errors in data transmission. It is Cyclic Redundancy Code (CRC) of length 32 bits. It is calculated for all the fields including the header and is calculated using a 32 degree polynomial.

Upon receiving the CTS packet, the sender can initiate data transmission to the receiver. If the data is successfully received by the receiver, then the receiver sends an acknowledgement to the sender as shown in Figure 3. 5.

3.3 Classification of MAC protocols

Traditionally, there are four different channel access schemes: Time division multiple access (TDMA), Frequency division multiple access (FDMA), Code division multiple access (CDMA), and Space division multiple access (SDMA) [Keoliver05]. In TDMA, all the nodes use the same frequency channel and each node is assigned with designated time slot(s) for data transmission. The nodes transmit in rapid succession, one after the other, each using its own time slot. Time synchronization among the nodes accessing the shared medium is required for the success of TDMA scheme. The technique that FDMA uses is similar to that of TDMA. The only difference is that instead of dividing the time, FDMA allocate different frequencies to each node. CDMA employs spread-spectrum technology and a special coding scheme to allow multiple users sharing the same physical channel where each node is assigned a unique code. SDMA on the other hand uses the spatial separation of the nodes for multiple channel access through spatial multiplexing and/or diversity. In general, different networking technologies may share access

via different methods or a combination of multiple methods such as TDMA, FDMA, CDMA, or SDMA.

In wireless networks, the medium access scheme can be distributed and centralized [Achandra00]. Based on the mode of operation wireless MAC protocols can also be broadly classified as random access protocol, guaranteed access protocol and hybrid access protocol. In random access MAC protocol, each node tries to access the transmission medium access in a random manner while in the guaranteed access MAC protocols, nodes access the transmission medium in a systematic manner by employing a master-slave procedure or sharing token to take turn. Hybrid protocols use a blend of guaranteed access and random access for accessing the transmission medium. Similarly, for resolving the challenges such as hidden terminal problem, resource constraints, and application requirements, researchers in the literature have investigated a number of MAC protocols specifically for WSNs through either extension of existing MAC protocols or proposing new medium access concepts. Similarly, based on the method used for contention avoidance, MAC protocols in WSNs can be roughly classified into three categories as following.

1. Contention based MAC protocols
2. Schedule based MAC protocols
3. Hybrid and Event based MAC protocols

3.4 Contention-based MAC Protocols

The contention-based MAC protocols allow multiple nodes to access the medium at the same time. Collisions may then occur, but are handled with different contention resolutions such as random backoff, RTS/CTS exchange and collision avoidance techniques. A classic example is

Carrier Sense Multiple Access (CSMA) in which a node will sense the medium for ongoing communication before attempting a message transmission. If the node determines that the medium is busy, it will back off and retry later. When the medium is sensed as clear, the node waits for a random period, the contention period, before sending. The contention period decreases the probability that two nodes start sending at the same moment and therefore reducing data collision. The scheme of RTS/CTS message exchange in IEEE 802.11 DCF and timeout are often combined with the unique features of WSN applications in contention-based MAC protocols to optimize the network performance such as energy consumption, lifetime, latency, or throughput. Examples of contention-based MAC protocols in WSNs are S-MAC [Wye02], T-MAC [Tfdam03], WiseMAC [Aelhoiydi04], DMAC [Glu04], DSMAC [Plin04], AC-MAC [Fli06] [Jai04] and so on. In the subsequent sections, we introduce the basic protocol design of S-MAC and T-MAC protocols.

3.4.1 Sensor Medium Access Control(S-MAC) [Wye02]

It is observed that energy in wireless sensor networks is wasted in multiple processes including *idle listening*, *data collisions*, *overhearing*, and *control overhead*.

The *idle listening* is a state where the sensor node waits for another node to possibly transmit the data to it. In many sensor network applications, if nothing is sensed, nodes are in idle mode for most of the time. However, traditional MAC protocols such as IEEE 802.11 or CDMA requires nodes listening to the channel for possible transmission. Study shows that idle listening

consumes 50–100% of the energy required for receiving [Stemm97]. However, in many WSN applications, the nodes stay in the idle state far longer than the communication state, which in fact consume a significant portion of the node's energy. Data collisions lead to the corruption of data in transmitted packet which has to be discarded and the follow-on retransmissions increase energy consumption as well as the network latency. Similarly, overhearing transmissions among other nodes and the control overhead can contribute the energy wastage within WSN nodes.

S-MAC protocol tends to reduce aforementioned energy wastage using periodic sleep and listen cycle while introducing some penalty on the per-hop latency of the data transmission. S-MAC protocol assumes that all the nodes are used for one application or a set of applications. Since the sensor nodes have one common application goal and there might be a situation where one node holds more information than other nodes. S-MAC applies the concept of message passing to allow the node holding more data access the channel longer which in fact preserves the important application level fairness rather than per-hop fairness.

Periodic Listen and Sleep

Since sensor nodes stay in the idle state quite often for many WSNs applications, S-MAC protocol introduces a set of sleep and wakeup states to reduce the energy wasted in idle listening. In the sleep state, the nodes turn off their communication devices (which contribute most to the energy consumption) and keep other components on. By following a schedule, the nodes move from the sleep state to the wakeup state, after certain time interval. The certain time interval depends on the application that the nodes are performing. In the wakeup state, the nodes turn on their communication devices and participate in necessary communication with other nodes.

To follow the schedule for sleep/wakeup and communicate with neighbors in time, the nodes in S-MAC protocol require the periodic synchronization among their neighbors. To avoid the time synchronization errors, S-MAC uses two techniques. First, all the timestamps used for synchronization are not absolute but are relative. Secondly, listen period is longer than the clock drift. In S-MAC protocol, the nodes are free to choose the listen/sleep schedules but it is preferred that the neighboring nodes should synchronize with each other in order to reduce control overhead because a node can communicate with another node only if both are in wakeup state. In other words, it is ideal for the neighborhood nodes to listen at the same time and go to sleep at the same time. However, in a multi-hop network as shown in Figure 3.8, not all neighboring nodes can synchronize together to follow the same listen/sleep schedule. For example, sensor nodes A and B follow the same sleep/listen schedule. Similarly, sensor nodes C and D follow the same sleep/listen schedule, which might be different from the schedule followed by A and B. Nodes exchange their schedules by broadcasting it to all its immediate neighbors. This ensures that all neighboring nodes can talk to each other even if they have different schedules

Figure 3.8 An example of four nodes network

In S-MAC protocol, a node wants to talk to a neighbor, the node must wait until the neighbor is listening (or in wakeup state). If more than one neighbor wants to talk with a node, the neighbors have to contend for the medium access when the node is in wakeup state. For this contention, the scheme of RTS/CTS exchange is adopted. The node who first sends out the RTS packet owns the right to access the medium and the receiver will reply with a CTS packet. Upon

receiving the CTS packet, the node can finish the data transmission and follow the schedule to sleep or listen.

Choosing and Maintaining Schedules

Each node should choose a schedule, before the periodic listen and sleep, and exchange the schedule with its neighbors. The schedule is stored in a table which contains the schedules of the neighboring nodes. Selection of the schedule and insertion of schedules of its neighbors is done as follows.

1. Every node listens for certain amount of period to the transmission channel. If the node does not receive schedule advertisement, the node randomly selects its own listen/ sleep schedule and broadcasts the schedule in a SYNC packet to its neighbors specifying that it moves into the sleep state after t seconds. Without a neighbor's schedule to follow, this node chooses its schedule independently and the node is called the synchronizer.
2. During the listen period, if a node receives a SYNC packet from its neighbor prior to randomly selecting its own schedule. The node will follow the schedule specified in the SYNC packet it received from the neighbor node. Such a node is called follower. Assume that the follower recognizes that the sender of the SYNC packet will go to sleep state in t seconds. After waiting for a random delay t_d seconds to avoid potential collision from other followers, the follower rebroadcasts the schedule and specifies that it moves into sleep state after t-t_d seconds.
3. If a node selects a schedule and then receives another different schedule from its neighbor, then the node stays in wakeup state by following both received neighbor's schedule and his original schedule.

Maintaining Synchronization

The synchronization among the nodes in the wireless sensor network is maintained by sending the SYNC packets. SYNC packets contain the address of the source node and the time of its next sleep. To remove the clock synchronization errors, the time of next sleep is not absolute but it is relative to the time of transmission of the SYNC packet which is approximately equal to the time of reception of the packet by the destination. The destination node will start the timer immediately after receiving the SYNC packet. When the timer expires, the node moves into the sleep state. To send data packets and SYNC packets, the wakeup period is divided into two parts. In the first part, the nodes receive the SYNC packets and the second part is for receiving the RTS packets. Each part is further divided into slots for carrier sense prior to channel access of SYNC or data packet transmission.

Each node periodically broadcasts its schedule in SYNC packets to its neighbors such that the new joined nodes can follow the same schedule. For the new joined nodes, the schedule selection process is the same procedure as described above. Before identifying itself as a synchronizer, the new joined node will set the initial listen period long enough to increase the probability of picking up a neighbor's schedule.

Collision and Overhearing Avoidance

To avoid multiple neighbors sending data to a node simultaneously, S-MAC adopts the RTS/CTS exchange as well as the virtual and physical carrier sense mechanisms, which is proved to be an effective approach to address the hidden terminal problem [Pkarn90,

Bharghavan93, IEEE07]. All the nodes initially should sense the carrier before initiating a data transmission. If source node senses the channel and concludes the channel is busy, the source node moves into sleep state. The source node wakes up again when the destination node is in wakeup state. S-MAC sends the broadcast packets such as the SYNC packets directly without employing RTS/CTS exchange. For unicast packets, the source and destination nodes follow RTS/CTS/Data/ACK sequence during the data transmission process. In addition, every data packet contains a field which indicates the remaining transmission time it needs. This is similar to the concept of network allocation vector (NAV) in IEEE 802.11. Hence, a node knows how long it has to keep silent or move back to sleep state (prior to accessing the channel) after receiving a packet destined to another node.

Figure 3.9 An example of overhearing avoidance [Wye02]

S-MAC protocol reduces the energy that is wasted in overhearing. S-MAC moves any node into the sleep state whenever the node hears a RTS or CTS packet. This is due to the fact that subsequent data and acknowledgement transmission will normally take much longer time. For example, in Figure 3.9, node C is sending the data to node D. It is clear that node D and node C should not be on sleep state. Since the collisions occur on the receiver's end, the node E cannot send data and should be in sleep state in order to avoid collisions at node D. Node B theoretically can send data to node A while being in wakeup state since node D is not in node B's transmission range. However, node B cannot get any reply from node A and node B's transmission could cause collisions at node C when node C tries to receive the

acknowledgement. Similarly, node E cannot participate in the data transmissions when node C is talking with node D. Hence, all immediate neighbors of both the sender and the receiver should sleep after they hear the RTS or CTS packet. In other words, based on the NAV information carried in the RTS/CTS packet, the node can sleep to avoid overhearing until the current transmission is over.

Message Passing

A message is a meaningful collection of data which can be one large packet or a series of short packets. On one hand, when a long message embedded in one packet is corrupted message, the retransmission is costly in terms of energy consumption, latency, and bandwidth utilization. On the other hand, transmission of a long message by using of multiple short and independent packet results in significant control overhead such as the RTS/CTS exchange. Hence, S-MAC protocol fragments the long message into many small fragments, and transmits them in burst. Only one RTS/CTS exchange is employed for the whole burst to reserve the medium for transmitting all the fragments. The transmission of a data fragment is assumed to be successful only if the sender receives the ACK from the receiver. If the sender does not receive the ACK packet, it will extend the reserved transmission time for one more fragment, and re-transmit the current fragment immediately. The acknowledgement packet is used after receiving each data fragment, in order to overcome the hidden terminal problem. The NAV information of current transmission is also present in the acknowledgement and data packets. In this way, a node in the path could know about the remaining duration for the ongoing data transmission even when there are corrupted packets or the node wakes up in the middle of the data transmission.

Energy Saving vs. Increased Latency

To analyze the delay penalty introduced by S-MAC, let us first take a look at the delays which are inherent to contention-bested MAC protocols (e.g., IEEE 802.11 DCF) in a multi-hop network. The delays include carrier sense delay, backoff delay, transmission delay, propagation delay, processing delay and queuing delay. However, S-MAC introduces an extra delay called *sleep delay*, which is experienced by the source when it finds the intended destination in sleep state. In this case, the source node needs to wait until the destination node changing to wakeup state. Assume a frame is defined as a complete cycle of listen and sleep. Then the average sleep delay will be as shown in Equation (3.2) when the data packet arrives with equal probability during a frame.

$$D_s = \frac{T_{frame}}{2} \tag{3.2}$$

Where D_s denotes the sleep delay and T_{frame} denotes the time frame and is a summary of T_{listen} denoting the time period of listen state and T_{sleep} denoting the time period of sleep state as shown in Equation (3.3).

$$T_{frame} = T_{sleep} + T_{listen} \tag{3.3}$$

The relative energy savings from S-MAC is given by Equation (3.4), where the last item is the duty cycle of the node. It can be seen that the smaller the listen period, the shorter the average sleep delay is.

$$E_s = \frac{T_{sleep}}{T_{frame}} = \frac{T_{frame} - T_{listen}}{T_{frame}} = 1 - \frac{T_{listen}}{T_{frame}}$$

(3.4)

S-MAC protocol reduces the energy consumption of the nodes thereby, increasing the lifetime of the entire network. But S-MAC introduces some delay in order to lessen the energy consumption. Thus, it is not good idea to apply S-MAC for MAC in WSN which is used for time critical applications,

Evaluating S-MAC Protocol

Experiments in [Wye02] show that S-MAC has very good energy conserving properties when compared to that of IEEE 802.11 DCF. On a source node, an IEEE 802.11-like MAC consumes 2–6 times more energy than S-MAC for traffic load with messages sent every 1–10s. To reduce the latency in S-MAC, a new technique called *adaptive listen* is introduced in [Wye04]. The basic idea is to switch the nodes from the low-duty-cycle mode to a more active mode. Specifically, adaptive listen 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 ongoing transmission. Rather than waiting for the scheduled listen time, the wakeup node can immediately receive data from the neighbor if it is the next-hop node. Otherwise, the node will go back to sleep until its next scheduled listen time.

3.4.2 Timeout MAC (T-MAC) [Tvdam03]

To further resolve the problem of idle listening in a WSN, T-MAC is proposed as another contention-based MAC protocol to reduce energy consumption through turning off radio components of the nodes when they are not needed [Tvdam03]. The basic idea of T-MAC is to turn on node radio at synchronized time and turn them off after a certain time-out when no communication occurs during some time. Unlike its predecessor S-MAC which turns on the radio according to a predefined schedule, T-MAC dynamically adapts a listen/sleep duty cycle in a different way through fine-grained timeouts. As a result, the T-MAC protocol can save more energy than the S-MAC in a network where message rates vary.

Protocol Design

Similar to S-MAC protocol, T-MAC protocol also uses the periodic sleep and wakeup states to save energy in WSNs. In the sleep state, the node has the sensing devices turned on and the data sensed is put into the queues. The node in sleep state also accepts new messages from the neighboring nodes and these messages are queued. In the active or wakeup state, the nodes keep listening and transmitting data as needed. For the data transmission in the active state, T-MAC adopts the Request-To-Send (RTS), Clear- To-Send (CTS), Data, Acknowledgement (ACK) scheme, to provide collision avoidance and reliable transmission. A node transits from active state to sleep state when no *activation event* occurs within a time span of TA. An activation event is defined as one of the follows.

1. the expiration of a periodic frame timer
2. the reception of any data on the radio

2. the sensing of communication on the radio
3. the end-of-transmission of a node's own data or acknowledgement packet
4. the knowledge (obtained through overhearing prior RTS and CTS packets) that a data exchange of a neighbor has ended

The minimal amount of idle listening per frame is determined by the value of TA. Since messages received in sleep state must be buffered, the maximum frame time is bounded by the buffer capacity.

Clustering and Synchronization

The synchronization in T-MAC protocol is done using a technique called virtual clusters [Wye02]. In virtual clustering, nodes with same schedule form clusters, without enforcing the same schedule to all nodes in the network. Virtual clustering allows the node to broadcast the schedule and anticipates that the node maintains the schedules of their neighboring nodes. Initially, every node starts its operation by listening and waiting. If a node receives nothing after listening and waiting for certain amount of time, it chooses a frame schedule and broadcasts its SYNC packet to the neighbors. On the other hand, if the node receives a SYNC packet from any one of the neighbors, then the node follows the same schedule in the SYNC packet it received. Furthermore, if the node receives a SYNC packet after broadcasting its own SYNC packet, then it follows both schedules and notifies the sender of the SYNC packet that there exists more than one schedule. Nodes broadcast their schedules once in a while. In irregular intervals, the nodes listen for complete time frames, so that the nodes could detect different schedule that exist in the same cluster. The nodes should transmit the data at the start of the active state since the neighbor

nodes within the virtual cluster (with the same schedule) and neighbors that have adopted the schedule as extra, are awaken in the active state.

Contention Resolution

In T-MAC, a frame consists of active state and sleep state. In the sleep state, the sensed data are queued and stored in queues to be transmitted. Hence, at the beginning of active state in a frame, each node may have buffered a large amount of data in the form of data bust (to be sent out). This results in higher contention for the medium access at the beginning of the active state. RTS/CTS exchange is employed in T-MAC for the channel contention. A node keeps sensing the medium for a random time with a fixed contention interval before sending a RTS packet. After sending out a RTS packet, the sender may not receive a CTS reply if the receiver is in sleep state. Even in active state, the receiver may not be able to send CTS reply if the RTS packet is lost due to collision or the receiver is prohibited from replying due to an overheard RTS or CTS. Since the receiver could be in active state, it makes sense for the sender to retry the RTS transmission. The sender will go to sleep if there is still no CTS reply after two retires.

As mentioned earlier, the active state ends when no activation even has occurred for a period of *TA*, which means the sender will automatically transit to the sleep state if the sender does not receive the CTS packet in time. Therefore, the value of TA must be selected such that the sender is able to receive the CTS reply [Tvdam03]. For a third neighbor node who overhearing the RTS or CTS packet, unlike S-MAC which requires the node going to sleep state, T-MAC makes overhearing as an option. The argument is that the overhearing avoidance could dramatically decrease the throughput performance of WSNs since it is very possible the node that overhears

the RTS/CTS is the receiver of subsequent message.

Early Sleeping in T-MAC

The research in [Tvdam03] found that T-MAC does not perform well when all the nodes send the data to a data sink. For example, assume there are four sensor nodes A, B, C and, D. and the messages flow only in one direction: A→ B→ C→ D, as shown in Figure 3.10.

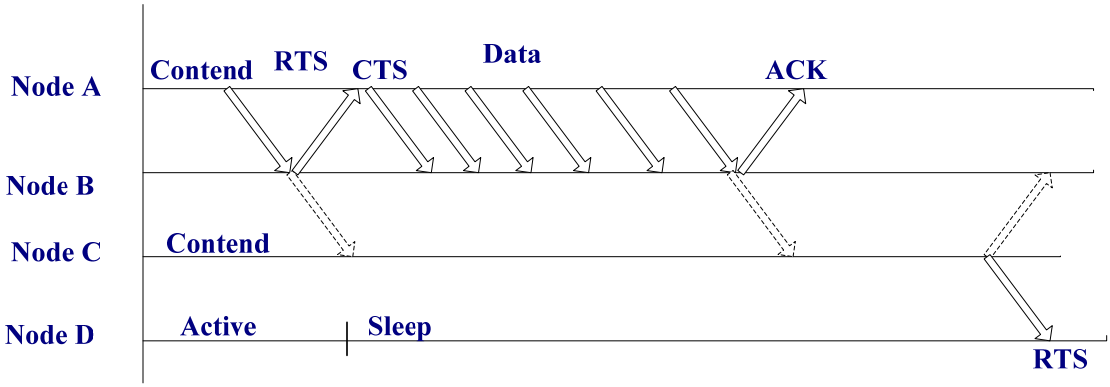


Figure 3.10 Early Sleeping Problem [Tvdam03]

In order to communicate with node D, node C has to contend for the transmission channel. Node C may lose the contention of the transmission channel to node A or node B. If node C loses the contention due to a RTS packet from node B, node C shall send a CTS reply to B, which will be overheard by node D. Accordingly node D can anticipate itself as the subsequent receiver and wake up when the communication between C and B is over. However, C must remain silent if node C loses the contention due to overhearing the CTS packet from B to A. In this case node D, who is totally blind to the communication between A and B, will go to sleep after the expiration of the TA timer. Hence, even in next contention round, node C wins the contention, node C

cannot talk with node D who is in sleep state. This observed behavior is called *early sleeping problem* since a node moves to sleep state even though a neighbor intends to communicate with it. There are two possible solutions for *early sleeping problem*: future request-to-send and taking priority on full buffers.

Future request-to-send

The basic idea of *future request-to-send* is to inform another node there will be a message for it even though at the current time the transmission medium is not available. The operation of FRTS is shown in the Figure 3.11. Once node C overhears the CTS packet from node B to node A, node C can immediately transmit a special packet called future request-to-send (FRTS) packet to node D if node C has data for node D. The FRTS packet contains the destination of FRTS packet as well as the information of the length of the ongoing data transmission which prevents node C sending data to node D. A node should not send FRTS packet if it is prohibited from data transmission.

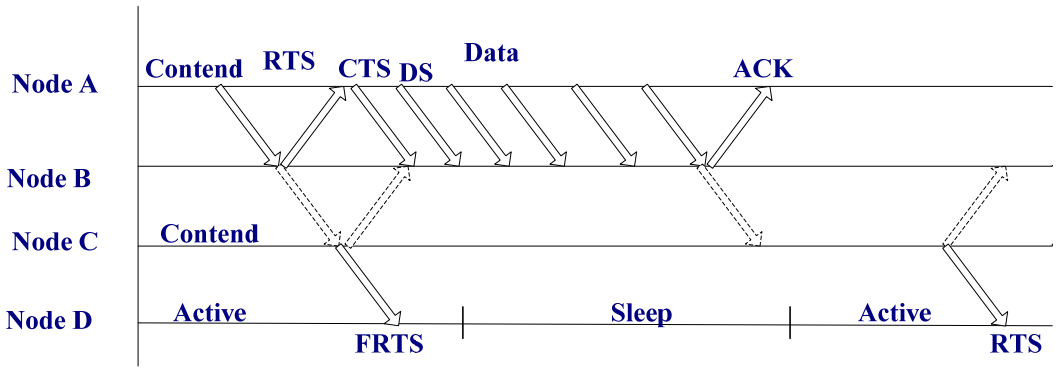


Figure 3.11 Future request-to-send [Tvdam03]

The destination node of FRTS packet must be in active state or wake up to receive data from the

sender of the FRTS packet when the ongoing communication is done. The destination node gets this information from FRTS packet. To prevent another node from occupying the transmission medium, the winner of previous contention (i.e., node A) sends a small dummy Data-Send (DS) packet prior to sending a burst of data. The DS packet contains no useful information. Hence, the collision between the DS packet and the FRTS packet does not affect the following data transmission.

Taking priority on full buffers

The second solution is based on the observation that a node may prefer sending to receiving when the node's transmit/routing buffers are almost full. As shown in Figure 3.12, assume that node B sends the RTS packet to node C, whose buffers are almost full. Instead of sending CTS reply to node B, node C initiates a data transfer with node D by sending RTS packet to node D. However, with this scheme, the node who is the intended receiver of prior contention winner has a higher probability to control the transmission medium. In this example, node C loses the contention with node B. But luckily, node C is the winner's receiver and node C (not node B) actually owns the transmission medium now. Obviously, node D will not have the *early sleeping problem* in this case. In addition, *the full-buffer priority scheme* introduces a limited form of flow control into the network, which actually is useful for many nodes-to-sink communication scenarios in WSNs.

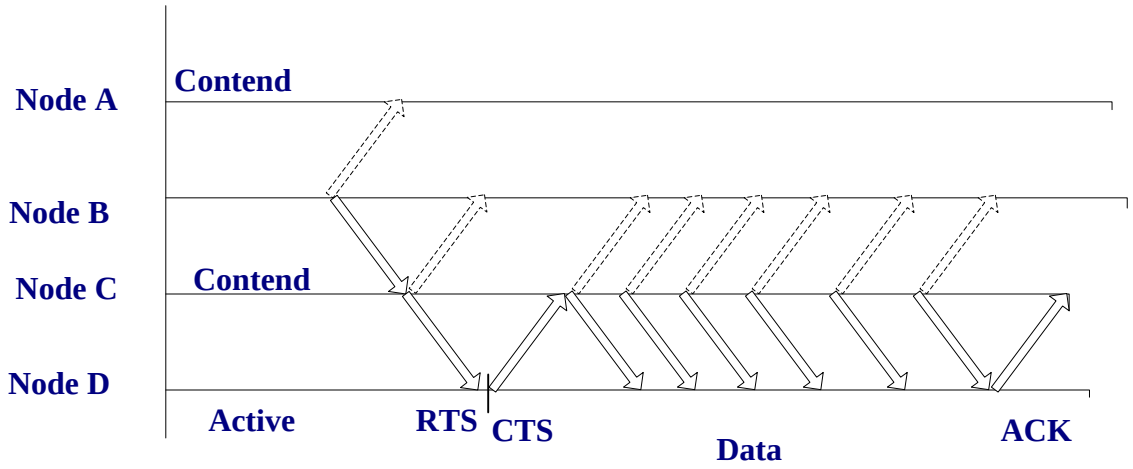


Figure 3.12 Taking Priority on full-buffers [Tvdam03]

However, when the high-load traffic is not flowed in a nodes-to-sink communication pattern, the data can flow this scheme must be applied carefully. The probability of collisions increases rapidly when, the nodes in a random communication pattern start taking priority. These collisions reduce the overall performance of the WSNs network. Therefore, T-MAC uses a threshold to limit nodes taking priority on full-buffers.

Evaluating T-MAC Protocol

T-MAC introduces the concept of turning off the radio when a certain time-out occurs, which presents an effective way to address the idle listening problem and decreases the energy consumption in a volatile environment where the message rate fluctuates, either in time or in location [Tvdam03]. Simulations show that the T-MAC protocol can save as much as 96% of the energy compared to a traditional CSMA-based protocol by using the radios for as little as 2.5% under a very low traffic load. With a high traffic node, T-MAC protocol does not increase

the latency and ensures a high throughput through not entering the sleep state. Under homogeneous traffic load, T-MAC and S-MAC achieve similar reductions in energy consumption (up to 98%) compared to the CSMA protocol. However, in network where message rates vary, the T-MAC protocol saves more energy than its predecessor S-MAC which only turns on the node radio for a fixed period.

T-MAC protocol reduces the energy consumption of the nodes thereby increasing the lifetime of the network without introducing any latency. T-MAC reduces the time required for the transmission of the data from the source node to the destination node. T-MAC protocol also solves the early sleeping problem by introducing future request to send and taking priority on the buffers. By solving the problem of early sleeping, T-MAC protocol can be applied to network when the data flows from the nodes to the sink node.

3.5 Schedule-based MAC Protocols

In schedule-based medium access, each node uses the shared transmission media based on a schedule. Similar, to TDMA-based protocols, time normally is divided into so called time slots of fixed length. The schedule determines the assignment of the time slots in a way that conflicts do not exist and each node gets an opportunity to use the medium. Often, these schedules are repeated after a certain periods and the nodes form a cluster. Since each node can access the

shared medium only in the dedicated time slot, schedule-based MAC protocols generally can avoid contentions, collisions and idle listening. Without additional overhead, the schedule can also easily transit a node into sleep state for energy saving. In addition, QoS and priority support can be conveniently achieved with schedule-based MAC protocols. However, a number of challenging issues arise when design schedule-based medium access schemes for the resource-constrained WSNs.

1. High quality clock synchronization among the nodes is not easy to achieve.
2. The dynamics of WSNs including nodes addition, nodes failure and mobility make effective slot assignment difficult.
3. Slot assignment in multi-hop WSNs is challenging.
4. Poor scalability and complexity in the schedule maintenance may significantly degrade the network performance.

In the literature, a number of studies have been conducted to design efficient schedule-based medium access schemes while resolving the aforementioned challenges. Examples of schedule-based MAC protocols are TRAMA [Vrajendran06], LEACH [Bwendi02], SMACS [Ksohrabi00], FLAMA [Vrajendran05], SPARE MAC [Lcampelli07], μ-MAC [Abarroso05], VTS MAC [Eelopez06], ER-MAC [Rkannan03], BMA MAC [Jli04], etc. For example, the low-energy adaptive clustering hierarchy (LEACH) protocol introduces the concept of hierarchy into WSNs for transferring data from the sensor nodes to the base station while the flow-aware medium access (FLAMA) protocol uses distributed election, by using the information of the flow, two-hop neighborhood, and simple traffic adaptive scheme for energy efficient channel access. In slot periodic assignment of reception (SPARE MAC) protocol, the nodes, which are

receivers at a particular instance of time, receive the reception schedule and propagate the information of reception schedule to all the neighbors. The μ-MAC protocol divides the transmission channel into contention and contention-free period and relies on the information provided by upper layers. The virtual time division medium access (VTS-MAC) protocol divides the nodes into clusters. In VTS-MAC, the time line is divided into time slots such that the number of nodes in the network is equal to the number of time slots. On the other hand, the bit map assisted MAC (BMA MAC) protocol proposes an intra cluster MAC protocol which divides the nodes in the network into clusters. The nodes in the cluster can communicate to the cluster only when there is an occurrence of significant events.

In the rest of this section, we particularly introduce the basic idea of the Traffic Adaptive Medium Access (TRAMA) protocol [Vrajendran06].

3.5.1 TRAMA (Traffic Adaptive Medium Access) [Vrajendran06] Protocol

TRAMA protocol is a schedule based MAC protocol for WSNs which saves the energy by making sure that there will be no collisions in the data transmission and by making the nodes enter a low-power state whenever, the nodes are not the intended receivers or intended transmitters. TRAMA protocol uses an adaptive selection for electing the nodes which transmit at a particular period of time and allows nodes to determine when they can transit into sleep mode. With traffic information, TRAMA can avoid assigning time slots to nodes with no traffic to send.

The time line of TRAMA protocol is shown in Figure 3.13, which includes *random access* and

scheduled access slots. The random access slots are called signaling period and the scheduled access slots are called transmission period. During the signaling period, the nodes broadcast the one-hop neighborhood information among neighboring nodes such that each node can obtain two-hop topology information around itself. During the transmission access slots, the nodes transmit the data and propagate schedule for contention-free data exchange. The schedule information consists of the set of receivers for the traffic originating at the node and TRAMA assumes that the clock synchronization is done previously. In the contention-free period, time is divided into small time slots and the schedule is fixed. When the contention-free period is over, the nodes fall back to the random-access period. The lengths of the signaling time slots and the transmission time slots depend on the type of application. Signaling time slots will occur more often for the dynamic scenarios, where the nodes move from one location in the network to other locations in the network. On the other hand, for the static scenarios where the nodes do not have mobility, the signaling time slot will be shorter. Since in wireless sensor networks, the sensor node does not move very often from one location to another location, the signaling time slot will be shorter.

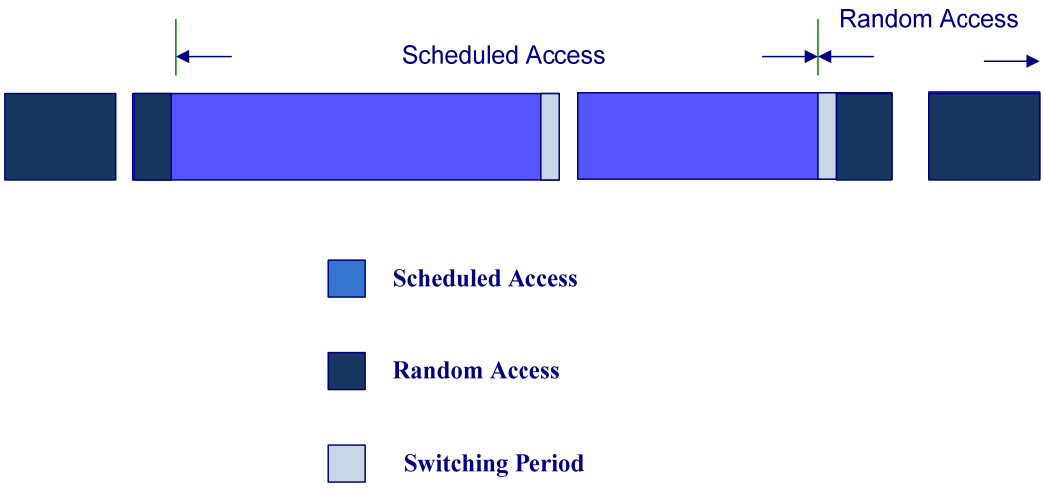


Figure 3.13 Time line of TRAMA protocol [Vrajendran06]

TRAMA consists of three components, which are

1. Neighbor Protocol
2. Adaptive Election Algorithm
3. Schedule Exchange Protocol

Neighbor Protocol

TRAMA protocol starts its operation in the signaling period. In the signaling period, every node chooses a random time slot and broadcasts the one-hop neighbor information among neighboring nodes. At the end of the signaling period, it is expected that all the nodes can discover their neighbors. Hence the main purpose of the signaling period is to permit node additions and deletions as such that the changes in topology can be discovered. The connectivity information in the network is found by these signaling packets. Figure 3.14 shows the format of the header of the signaling packets. Signaling packets carry incremental neighborhood updates and if there are no updates, signaling packets are sent as "keep-alive" beacons. Otherwise, if a node is not heard for a certain period of time, the node is assumed being disconnected from the network. The incremental updates from a node include the one-hop neighborhood information of this node in terms of added and deleted neighbors.

Type	Source Address	Destination Address	Delete Number	Add Number	Deleted Node ID's	Added Node ID's

Figure 3.14 Signal Header [Vrajendran06]

Once all the one-hop neighbors of a node, say node B, send the corresponding one-hop information to B. Node B can earn all its neighbors' neighbors. In other words, node B

eventually will have all the information of B's two-hop neighbor nodes and can construct a two-hop local topology around node B.

Note that during the random access periods, signaling packets may be lost due to collisions, which can result in inconsistent neighborhood information across the network. To ensure consistent neighborhood information, the length of the random access period and the number of retransmissions of the signaling packets should be set according to the real network or application scenarios.

Adaptive Election Algorithm

After discovering the neighbors, TRAMA protocol employs the *adaptive election algorithm* to establish schedule. Nodes locally compute who is the absolute winner among the two-hop neighbors in certain time slots by calculating the priority function as Equation (3.5).

$$\text{prio}\,(u, t) = \text{hash}(u \oplus t) \tag{3.5}$$

In Equation (3.5), *u* is the node identification, *t* is the slot number, and *hash (…)* is a network-wide known hash function. Based upon the results of the priority function, time slots are reserved to the winner (i.e., node with highest priority). For energy efficiency, TRAMA switches nodes to sleep state whenever possible, and re-uses slots that are not used by the winner. For example, the winner may give up its transmission slot if it does not have any data to send and the slot could be used by another node.

At any given time slot *t* during the transmission period, the state of a node *u*, is determined

according to the two-hop neighbor information and the schedules announced by *u*'s one-hop neighbors. Each node has three possible states:

1. Sleep State
2. Receive State
3. Transmit State

A node is in the *transmit state* if the node has data to send and is the winner, i.e., has the highest priority based on the calculation in Equation (3.5). When a node is the intended receiver of the current sender, the node is in the *receive state*. Otherwise, the communication system of the node can be switched off and move into *sleep state*, since it does not participate any data exchange.

Schedule Exchange Protocol

The traffic-based schedule information is established and maintained by the *schedule exchange protocol*, which is further broadcasted among the neighboring nodes periodically during the transmission slots. The schedule is generated as follows.

Step 1: Each node computes the number of time slots required to transmit the data through the transmission channel, SCHEDULE_INTERVAL based on the rate at which packets are generated at this node.

Step 2: The node then pre-computes the number of slots in the interval $[t, t + \text{SCHEDULE\_INTERVAL}]$ for which the node will be selected as the transmitter. In other words, during that interval, the node has the highest priority among its two-hop neighbors and is assumed to be the winners of this interval.

Step 3: The node informs the intended receivers for these slots in order to avoid the collisions because all the neighboring nodes of the present node will have the information about the

transmission schedule of the present node.

However, if the node does not have data to send, the node marks the slots as VACANT and sends the information to the neighboring nodes so that other nodes can make use of the vacant slots. The last time slot in the winning interval is used for broadcasting the node's schedule for the next interval.

The nodes announce the schedule information by using schedule packets as shown in Figure 3.15. The schedule packet includes fields such as the *source address*, *timeout*, *width*, and *number of slots*, and *Bitmap*. The *source address* identifies who is announcing the schedule, the *timeout* indicates how long this schedule is valid, the *width* is the number of bits in the *Bitmap*, the *number of slots* is the total number of winning slots, and the *Bitmap* identifies the intended receivers. Since the data from MAC layer is targeting only at one-hop neighbors of the sender and the neighboring information is already provided by the neighbor protocol, there is no need to specify the receiver address in the schedule packet. Instead, TRAMA adopts a bitmap scheme to identify the intended receivers. The length of the bitmap is equal to the number of one-hop neighbors. Each bit in the bitmap represents a particular one-hop neighbor and the order is based on the IDs of the neighbor nodes. If the sender wants to send the data to a particular neighboring node, the sender will set the corresponding bit in the bitmap to 1. Otherwise, the corresponding bit is set 0 if the node is not the intended receiver. Hence, when all the bits in the bitmap are set to l, the schedule packet is a broadcasting packet since all the one-hop neighbors are the intended receivers. Similarly, multicast can be easily support by only setting the multicast group of bits to 1.

Figure 3.15 Schedule packet format [Vrajendran06]

A summary of a node's schedule is also sent with every data packet to minimize the impacts of loss in the schedule dissemination. Nodes maintain the schedule information for all the one-hop neighbors. The information is consulted when a node needs to decide where transmitting or giving up the slot. The updated schedule based on this decision will be carried by the summary within the data packet.

Performance Evaluation of TRAMA

TRAMA assumes that time is slotted and uses a distributed election scheme based on information about traffic at each node to determine which node can access the channel for transmission at any particular time slot. With the traffic information, TRAMA avoids assigning time slots to nodes with no traffic to send, and also allows nodes to determine when they can switch to sleep mode. The TRAMA protocol ensures a distance of three hops or more can concurrently transmit data. The performance of TRAMA depends mainly on the traffic pattern while the performance of S-MAC depends on duty cycle. Simulations in [Vrajendran06] show that TRAMA outperforms contention-based protocols (CSMA, 802.11 and S-MAC) in terms of energy consumption and throughput. However, TRAMA experiences a higher delay than the static scheduled-access protocols (e.g., [Bao01]) due to the scheduling overhead. Similar, to

TDMA-based protocols, TRAMA is well suited for sensor applications periodic data collection and monitoring, which are not delay sensitive but require high delivery guarantees and energy efficiency.

3. 6 Event-based and Hybrid MAC Protocols

There are also a number of MAC protocols, which are neither solely based on schedule nor contention (for example, [Ksarvakar08], [Ngajaweera08], [Kjamieson03], [Szhou07], [Jpolastre04], [Irhee08]), developed for wireless sensor networks in the literature. Some MAC protocols use a hybrid of contention-based and schedule-based concepts and some are event-based. Examples of hybrid and event-based MAC protocols are Zebra MAC [Irhee08], Sift MAC [Kjamieson03], FAMA/TDMA Hybrid MAC protocol [Ngajaweera08], EZ-MAC [Ksarvakar08], A^2-MAC [Szhou07], etc. In this section, we introduce examples of event-based protocols followed by a hybrid MAC protocol developed for WSNs.

FAMA/TDMA (Floor Acquisition Multiple Access/Time Division Multiple Access) hybrid protocol combines both FAMA and TDMA for providing medium access to all the nodes in the network. Initially the nodes in network contend for gaining the access to the transmission by sending RTS packets to the base station. The node with first successful RTS packet is given absolute access to the transmission channel to transmit the sensed data. In EZ-MAC (Utilized ZigBee MAC) protocol, the data is sent with low service access delay keeping access blocking ratio low by optimized structural sequence. It also uses scheduling scheme for WSNs. A^2-MAC (Application adaptive medium access control) is data collection protocol. It is hybrid slotted CSMA/TDMA protocol

3.6.1 Sift Medium Access Control [Kjamieson03]

In many WSN applications, the purpose of the sensor nodes is to detect events and report to a specific node called base station. Whenever an event occurs, all the nodes that sense the event will start transmitting the details of the event to the base station. Since multiple nodes that detect event very possibly are within a short distance and therefore share the same transmission medium. When all the nodes report at the same time, there will be contention in the transmission channel. Such a situation is known as spatially *correlated contention*. However, since multiple nodes detect the same event and may report similar sensed data to the base station, it is not necessary that all the sensor nodes report the event that has been detected. The event would be reported to the base station even if only a subset of the sensor nodes in the event's neighborhood actually reports the event. On the other hand, sensor nodes in the WSNs may fail or die due to battery or other causes and the density of the sensor nodes in a particular geographical area varies. Thus, it is desirable the MAC protocols for such wireless sensor networks can effectively handle the *correlated contention* along with time-varying *density*, which is the goal of the Sift MAC protocol [Kjamieson03].

Protocol Design

Similar to the traditional CSMA protocols, Sift MAC uses a contention window of fixed size of length 32 slots. The difference between the CSMA protocols and Sift MAC is that the probability of picking a slot in Sift MAC in a given interval is not uniform. In Sift MAC, nodes compete to transmit the data in slot $r \in [1, CW]$, where CW is the length of the contention window. The nodes compete for a particular slot based on the shared belief on the current living population

size, *N*, which changes after every slot in which no transmission occurs. The believed population starts off at some large value, indicating a correspondingly small per-node probability of winning the channel access. If no node transmits in the first slot, then each node sensing the medium reduces the believed number of competing nodes by multiplicatively increasing its transmission probability for the next slot. This process is repeated to enable the winner to be chosen rapidly across a wide range of potential population sizes without incurring long latency due to collisions. For example, if only one node competes for the transmission medium and then it gains the access in a particular slot of contention window to transmit the data. After the completion of the data transmission all the nodes compete for the new slots to transmit the data and estimate the values of *N*

Backoff probability distribution in Sift MAC

Assume every node picks up a slot $r \in [1, CW]$ using a non uniform probability function p_r. A slot $r \in [1, CW]$ in the contention window is said to be silent if no node chooses to transmit the data in that slot. Similarly, it is said that a slot $r \in [1, CW]$ has a collision if more than one node chooses the same slot. A sensor node can win a slot in the contention window only if one node chooses slot *r* for data transmission, which means slot *r* is the first non silent slot in the contention window. We call it a success if some sensor nodes win some slots. Sift MAC uses an increasing and truncated geometric distribution as in Equation (3.6) for the not uniform probability process p_r.

$$p_r = \frac{(1-\alpha)\alpha^{CW}\alpha^{-r}}{1-\alpha^{CW}} \text{ For } r \in [1, CW] \tag{3.6}$$

In Equation (3.6), α is a distribution parameter in the range of (0,1), which results in the

exponential increase of p_r. This means the later slots in the contention windows have higher probability.

Each station's choice of which, slot to pick can be viewed as a decision procedure with *CW* stages. A sensor node starts in stage 1 by estimating the value of the current living population N as N_1 and then chooses the slot 1 in contention window with same probability. If no node chooses slot 1 then each node assumes that the estimation was not correct and modifies the estimated value by decreasing the estimated value from N_1 to N_2. And then the node chooses the slot 2 with certain probability. If slot 2 turns to be another silent slot, then the estimated value is further reduced to N_3 and the above process is continued. Let N_r be one of the estimated value of N which is updated belief after having *r-1* number of silent slots in the contention window.

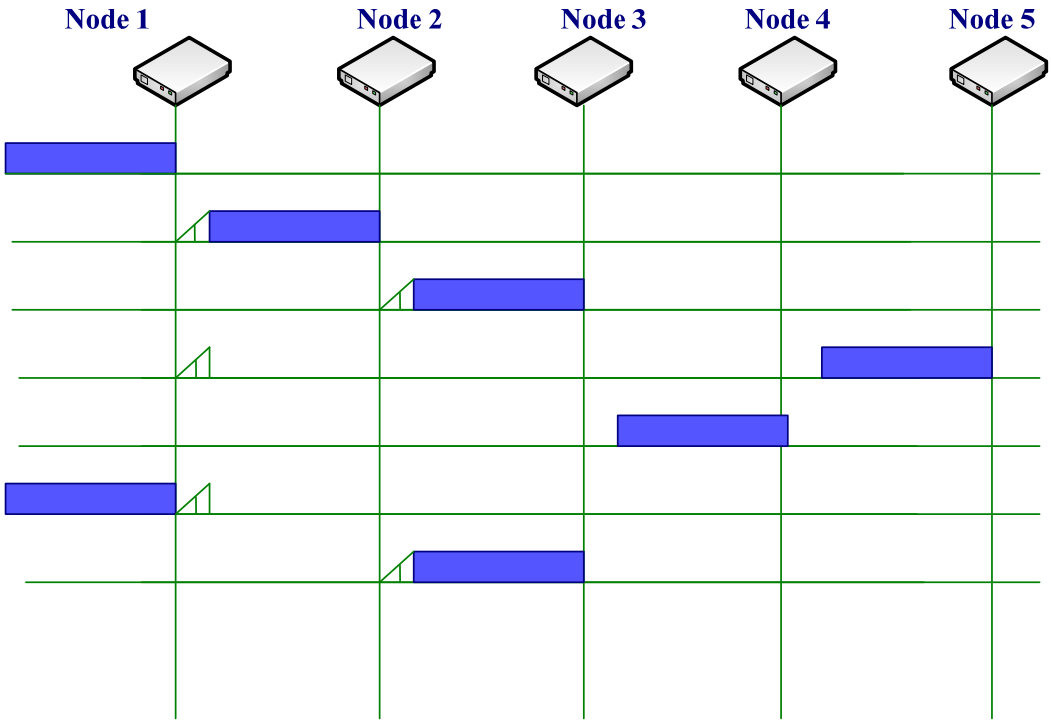


Figure 3.16 A timeline of five nodes running the Sift protocol. Every complete shaded region represents that the transmission medium is idle at that particular period of time. The shaded

regions represent that there is contention at that particular period of time. [Kjamieson03]

Since current living population $N \in [1, N_1]$, the probability of success should be kept constantly high in the decision process. Thus the following two properties should be held [Kjamieson03]:

1. The probability of success should be high when $N=N_1$
2. The probability of success should be constant

Assume that there are *r* silent slots in the contention window. Let p_r^1 be the probability that a node chose a slot *r* while there are *r-1* silent slots. Then the probability that slot *r+1* is a success slot is given by the Equation (3.7)

$$N_r\, p_r^1 (1 - p_r^1)^{N_r - 1} \approx N_r\, p_r^1 e^{-N_r\, p_r^1} \quad \text{(for large } N_r \text{ and small } p_r^1\text{)} \tag{3.7}$$

Thus the property (2) holds good only when $N_r\, p_r^1$ almost remains constant, so that the probability of success$N_r\, p_r^1 e^{-N_r\, p_r^1}$ does not vary significantly along with time.

To indentify a distribution that yields a constant $N_r\, p_r^1$, an exponential scheme is chosen as in Equation (3.8) for covering large N while only small number of slots in the contention window exists by which belief of the population reduces.

$$\beta = \frac{N_{r+1}}{N_r} \tag{3.8}$$

In Equation (3.8), β is constant and is given by $0 < \beta < 1$. Assuming that there will be no collisions or no two sensor nodes choose the same slot in the contention window, then for sensor node *S*,

$$p_r^1 = P_r(S\ chooses\ r | silence\ in\ earlier\ slots)$$

$$= P_r(S\ chooses\ r | S\ did\ not\ choose\ earlier\ slots)$$

$$= \frac{P_r(S\ chooses\ r)}{P_r(S\ did\ not\ choose\ earlier\ slots)}$$

$$= \frac{p_r}{1 - (p_1 + p_2 + \cdots + p_{r-1})}$$

(3.9)

$$= \frac{(1-\alpha)\alpha^{CW-r}}{1-\alpha^{CW-r+1}}$$

(3.10)

$$\frac{p_r^1}{p_{r+1}^1} = \frac{(1-\alpha)\alpha^{CW-r}}{1-\alpha^{CW-r+1}}\ \alpha \approx \alpha \ \text{(for small For small } \alpha^{CW-r}\text{)}$$

(3.11)

If the values of α and β are equated then Equations (3.8) and (3.11) can be equated, which results in

$$N_{r+1}\, p_{r+1}^1 \approx N_r\, p_r^1$$

This proves that the probability of success should be constant even when the value of N changes from N_1 to 1. As in property (1), the probability of success is high if N=N_1. Equation (3.10) also implies that $p_{CW}^1 = 1$, so if all the slots in contention window are silent and the last slot must be chosen by a node. Therefore, α should be chosen such that a node in stage *CW* believes that it is the only active node. This can be further illustrated by setting the value of current living population as 1which implies that $N = 1$.

From Equation (3.9), when $\alpha = \beta$ and $1 = N_{CW} = \alpha^{CW-1}N_1$. Thus the value of α is given by$\alpha = N_1^{\frac{-1}{CW-1}}$.

Protocol Specification

In Sift MAC, every node has four states as follows:

1. Idle State: in which a node waits for the data that is sent from other nodes.
2. Contend State: in which a node contends for the transmission channel and tries to gain the access of the transmission medium.
3. Receive State: in which a node receives the data that is sent from another node.
4. AckWait State: in which a node waits for the acknowledgement from another node after sending data out to the node.

The pseudo code for the transitions between the states is given in Figure 3.17. In Figure 3.17, the function *pickslot()* is used for picking up a slot for the transmission of the data using the Sift distribution specified in Equation (3.6). The directive *moveto(state)* changes the state of a particular node from the present state to the given state. The directive *wait(time)* waits for the period of time that is specified in the parameter.

Idle State
wait (channel idle)
if (recv frame for self)
moveto Receive
end if
if (xmit queue not empty)
moveto Contend
end if
Contend state
slot `pickslot()`
wait t_{difs}+ slot*t_{slot}
if (channel busy)
moveto Idle
end if
Transmit frame
moveto AckWait
Receive state
Check frame CRC
wait t_{sifs}
Send ACK
moveto Idle
AckWait state
wait $t_{ACK\ timeout}$
if (recv an ACK for self)
discard frame
moveto Idle
end if
Retransmit frame
moveto AckWait

Figure 3.17 Pseudocode for state transition in Sift MAC

t_{slot} is the minimal time separation such that if two nodes transmit more than t_{slot} seconds apart, the two nodes will hear the onset of each others' transmission. t_{sifs} is the amount of the time delayed at the beginning of a data acknowledgement packet for turning around from transmitting the packet to receiving the acknowledgement. t_{difs} is the amount of time delay added to the beginning of the data transmission of a new data transmission. Thus, t_{difs}+ slot*t_{slot} is the time taken for one complete data transmission and the subsequent transmission of acknowledgement. $t_{ACKtimeout}$ is the time for which a node waits to receive the acknowledgement.

Request to Send and Clear to Send Mechanisms

For avoiding the collisions, all the nodes in the sensor networks which implement the Sift MAC employ the RTS/CTS exchange scheme. In similar way Sift backoff distribution is used to compete on data packets, it can be used to compete on sending the RTS packet. Hence, one can just replace “frame” with “RTS” and “ACK” with “CTS” in the pseudocode, to achieve the RTS competition.

Performance Evaluation of Sift MAC

The basic idea of the Sift Mac is to use an increasing, non-uniform probability distribution within a fixed-size contention window, instead of using a time-varying contention window from which a node randomly picks a transmission slot as traditional contention-based MAC protocols. Sift MAC protocol is tuned for sensor networks where not every node has to report every detected event. Simulation studies show that the Sift MAC protocol performs well when spatially-correlated contention occurs and adapts well to changes in the active population size. In specific, results show that Sift MAC protocol improves over 802.11 in terms of report latency by up to a factor of 7 as the number of nodes reporting an event scales up to 512.

3.6.2 Berkeley Medium Access Control (B-MAC) [Jpolastre04]

To meet the requirements of wireless sensor network deployments and monitoring applications, B-MAC protocol is designed to achieve the following goals

1. Low power listening

2. Effective Collision Avoidance
3. Simple Implementation, Small Code and RAM size
4. Effective Channel Utilization at low and high data rates
5. Reconfigurable by the network protocols
6. Tolerant to changes in radio frequencies and network topology
7. Scalable to large number of nodes

B-MAC protocol provides certain interfaces for achieving these goals. These interfaces are listed in the Figure 3.18. For sensing the transmission channel, B-MAC protocol uses Clear Channel Assessment (CCA) and packet backoffs.

```
interface MacControl{
command result_t EnableCCA();
command result_t EnableCCA();
command result_t DisableCCA();
command result_t EnableAck();
command result_t DisableAck();
command void* HaltTx();
}
interface MacBackoff {
event uint16_t initialBackoff(void* msg);
event uint16_t congestionBackoff(void* msg);
}
interface LowPowerListening {
command result_t SetListeningMode(uint8_t mode);
command uint8_t GetListeningMode();
command result_t SetTransmitMode(uint8_t mode);
command uint8_t GetTransmitMode();
command result_t SetPreambleLength(uint16_t bytes);
command uint16_t GetPreambleLength();
command result_t SetCheckInterval(uint16_t ms);
command uint16_t GetCheckInterval();
}
```

Figure 3.18 Interfaces of B-MAC protocol [Jpolastre04]

Protocol Design

In B-MAC, signal strength is sampled when it is assumed that the transmission channel is free. For example, the transmission channel is free when the ongoing transmission is completed or when communication device is not receiving any data. The sampled data is entered into a queue. The median of the sampled data is found and is added to an exponentially weighted moving average with decay, α. The median is used for adding robustness to the noise floor estimate. After the noise floor is estimated the request for monitoring the received signal strength starts monitoring the transmission channel. B-MAC protocol searches for outliers in the received signal strength. For instance, if a node senses the outlier, it declares that the channel is unoccupied since a valid packet could never have an outlier below the noise floor. If there is no outlier found in the samples then it is concluded that the channel is busy.

Using the MacControl interface in Figure 3.18, nodes in B-MAC protocol can turn the CCA on or off. If CCA is disabled, scheduling protocol is implemented in B-MAC protocol. B-MAC protocol uses packet backoff when CCA is enabled. For the packet backoff, instead of setting a backoff time, B-MAC uses an event driven approach, which may return backoff time or ignore the event. If the event is ignored, small backoff time is set. After the initial back off time, CCA outlier algorithm is run. If the channel is not clear, the service for congestion back off time are signaled by the event.

B-MAC protocol also provides link-layer acknowledgement support. If the application requires the acknowledgement, then the acknowledgement is sent to the source node from the receiver node. After receiving the acknowledgement the source node sets a bit in sender's transmission

message buffer. B-MAC uses low power listening (LPL) for periodic transmission channel sampling. Every node in B-MAC protocol senses the transmission channel for activity in the transmission channel. If it senses an ongoing data transmission then it waits for the completion of the data transmission. After the data transmission the node moves into the sleep state. If no packet is received then a timer pushes the node into the sleep state. The interval between two LPL samples is maximized in order to minimize the time spent in sampling the transmission channel.

Performance Evaluation of B-MAC

B-MAC protocol performs better when compared to that of S-MAC and T-MAC in terms of throughput and energy consumption. The performance of S-MAC and T-MAC protocols is dependent on the length of the duty cycle.

In summary, B-MAC provides a flexible interface to obtain ultra low power operation, effective collision avoidance, and high channel utilization in wireless sensor networks. B-MAC effectively performs clear channel estimation. While supporting on-the-fly reconfiguration and provides bidirectional interfaces for system services, B-MAC employs an adaptive preamble sampling scheme to reduce duty cycle, minimize idle listening, and achieve low power operation. B-MAC may be configured to run at extremely low duty cycles and does not force applications to incur the overhead of synchronization and state maintenance like other MAC protocols. Experimental studies show that B-MAC's flexibility result in better packet delivery rates, throughput, latency, and energy consumption than S-MAC[Jpolastre04].

3.6.3 Zebra Medium Access Control (Z-MAC) [Irhee08]

Z-MAC protocol is a hybrid protocol which combines the merits of TDMA and CSMA while offsetting the demerits of both the schemes. Z-MAC uses CSMA at the base but follows TDMA depending on the contention level. The overhead of Z-MAC protocol is the setup phase, which is done at the beginning. In the setup phase, the nodes are assigned with the timeslots for the data transmission. The nodes use the assigned timeslots for the transmission of the sensed data in a particular period of time known as frame. A node is called the owner of a time slot if it wins the access of the transmission medium; otherwise the node is known as non-owner. The non-owners of the time slot have lower priority to transmit the data when compared to that of owners of the time slot. The priority is set using the contention window size. If at a particular point of time, the owners do not transmit the data, then the non-owners of the time slot may transmit the data by using the time slot that is left unused by the owner of the time slot. Z-MAC protocol performs similar to TDMA when the level of contention is low (or the traffic load is low) and it performs similar to CSMA when the level of contention is high (or the traffic load is high).

Z-MAC Setup Phase

Initially Z-MAC protocol runs the setup phase which consists of the following steps.

1. Neighbor Discovery
2. Slot Assignment
3. Local Frame exchange
4. Global time synchronization

1. Neighbor Discovery

In the step of *neighbor discovery*, every node in the network finds the one-hop neighborhood by sending *ping* messages, containing the current list of one-hop neighbors, to its one-hop neighbors. The two-hop neighborhood information then can be found by combining all the received one-hop neighborhood information of its neighbors.

2. Slot Assignment

In slot assignment, Z-MAC uses the Distributed RAND (DRAND) [Irhee06] algorithm to assign the time slots for data transmission. DRAND algorithm is the distributed implementation of RAND algorithm [Rramanathan97]. RAND algorithm is a centralized algorithm for assignment of time slots. DRAND algorithm runs in rounds. There are four states in DRAND which are IDLE state, REQUEST state, RELEASE state and GRANT state. The state diagram of DRAND is shown in Figure 3.19. Initially, every node is in IDLE state. During the IDLE state, the node tosses the coin for which the probability of getting a head or a tail is 1/2. If the result is a head, then it runs a lottery. If it loses the lottery then it remains in the same state. If it wins the lottery it moves into REQUEST state where the node broadcasts a request message to all its one-hop neighbors.

Consider node B which is one-hop neighbor to node A. If node B receives the REQUEST message from node A, when node B is in IDLE state or in RELEASE state then it responds with *grant* message and moves to GRANT state. If node B is in REQUEST state or GRANT state then it responds with a *reject* message to node A. If node A does not receive the *grant* message or *reject* message within the specific period of time, it sends the same request message again.

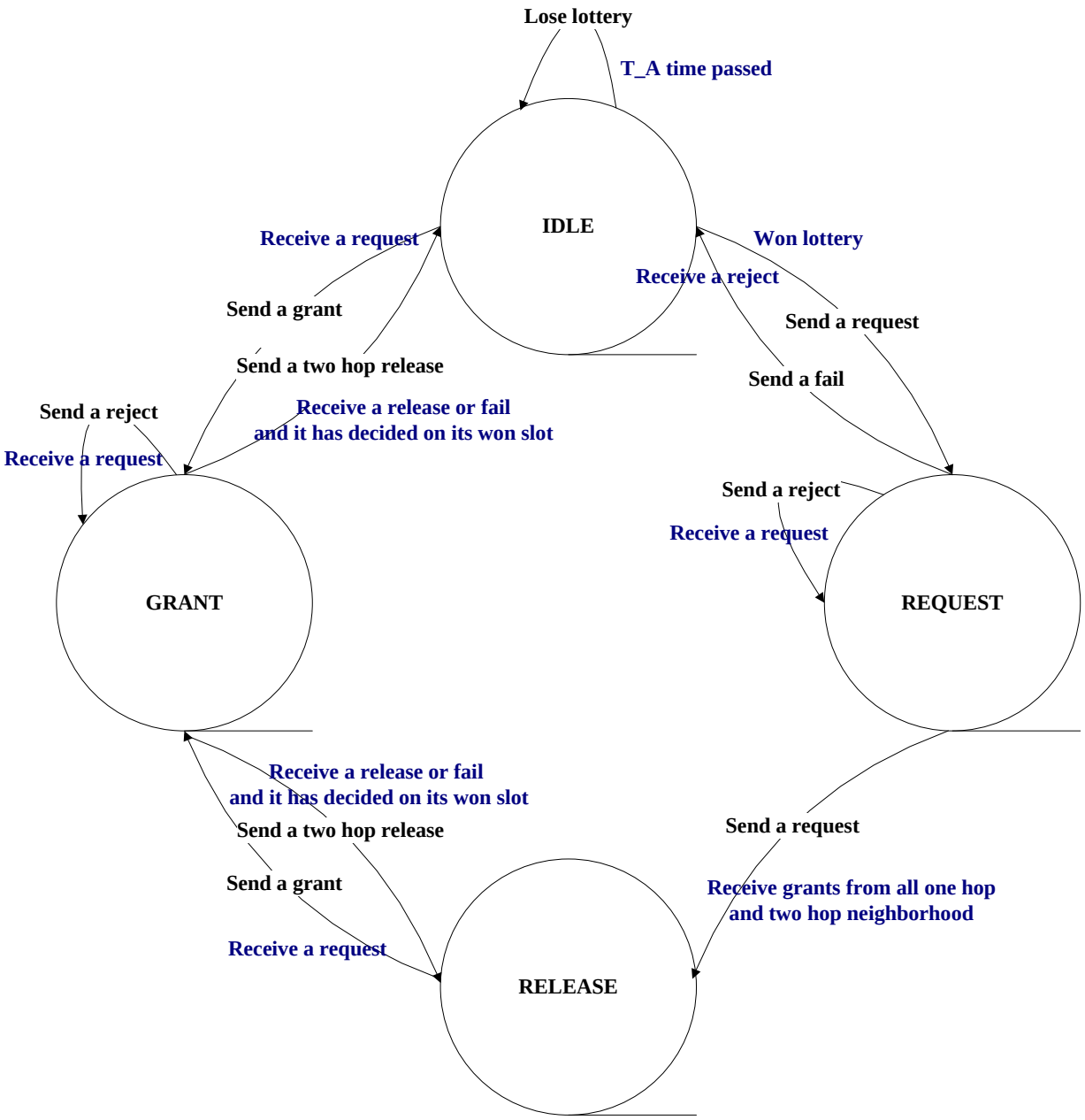


Figure 3.19 DRAND state diagram [Irhee06]

3. Local Framing

After the slot assignment, every node in network gets a time slot for transmitting its data to its intended destination. Then the node needs to decide on the period in which it can use the time

slot for transmitting the data to the intended destination. This period is called *the time frame* of the node. After the node decides the period for transmitting its data, the node should propagate the *maximum slot number* (MSN) to the entire network and adapt to local time slot changes. If there is any addition of new nodes in the network, then DRAND will assign the new time slots for new added nodes. The change in MSN should also be propagated to all nodes in the network.

Under high contention conditions, Z-MAC requires clock synchronization. Z-MAC protocol uses the Real time Transport Protocol (RTP/RCTP) [Hschulzrinne96] for clock synchronization under high contention conditions. In RTP/RCTP, every node in the network sends the control message at a rate that is limited to a small fraction of session bandwidth and every node in the session adjusts its bandwidth according to the allocated session bandwidth. In Z-MAC protocol, every node limits the data sending rate to predetermined data sending rate, which is determined based on the energy and bandwidth.

4. Global time synchronization

Local framing assumes that all the nodes are synchronized initially at the time slot 0. This could be achieved by fixing a predetermined time to synchronize the time slot 0. All the nodes are synchronized at slot 0 using Timing-sync protocol for WSN (TPSN) [Sganeriwal03]. TPSN assumes that every node has 16 bit register which acts as a clock that is triggered by the crystal oscillator. TPSN runs in two steps. In the first step, the nodes in the network construct a hierarchical structure. Every node *k* belongs to a level *i*. The nodes in level *i* can communicate with the nodes in level *i-1*. Only one node will be at level zero. This is called the root level. In the second step which is *synchronization* step, all the nodes at level *i* synchronize with the nodes

at level *i-1*. Thus every node in the network synchronizes with the root level. Thus all the nodes in the network get synchronized at slot zero. After the global time synchronization, each node in the network performs the local time synchronization.

Z-MAC protocol acts similar to TDMA when the contention level is lows and acts similar to CSMA when the contention level is high in the network. Z-MAC protocol requires the time synchronization only among the neighboring nodes and when there is high-level contention in WSNs. To implement the time synchronization among the neighboring nodes, Z-MAC protocol uses Real time Transport protocol [Hschulzrinne96] for local time synchronization.

Transmission Control of Z-MAC Protocol

Every node in Z-MAC protocol can be in any one of the following two modes.

1. Low Contention Level (LCL)
2. High Contention Level (HCL)

Every node will be in *Low Contention Level* (LCL) until it receives the *Explicit Contention Notification* (ECN) message from a two-hop neighbor node. A node sends ECN message if it experiences high contention. Once received a ECN message, the node transits to HCL mode.

Explicit Contention Notification (ECN) message

ECN messages intend to notify the two-hop neighbors of the current owner of the time slot not to act as hidden terminals when the contention level is high. In Z-MAC protocol, every node needs to decide the contention level based on the estimate of the contention level. The nodes can estimate the contention level using the following two methods.

1. To measure the packet loss in the acknowledgements

Since two-hop contention may result in collision and hence data loss, the source node could measure the contention level by measuring the packet loss in the transmission. Based on the received acknowledgement packets, a node could calculate the packet loss percentage and decide the contention level. However this technique requires the receiver to send acknowledgement back to the sender and incurs extra overhead and decreased channel utilization.

2. To measure the noise level of the channel

Whenever the contention level is high, the noise level in transmission channel increases. Measuring the noise level in the transmission channel does not require any extra overhead. For measuring the noise in the transmission channel, the nodes calculate the number of noise backoffs. A noise backoff is a backoff transmitted by the source node when it senses the transmission channel using Clear Channel Assignment (CCA). With CCA, a node in the network can transmit only when the node senses the channel to be clear. When the node experiences the contention, it takes the backoff message. When more than one destination take the backoff message, the node sends the one-hop ECN message to the node indicating a high contention level. If node *j* receives the ECN message sent by node *i*, node *j* first checks whether it is the destination of the ECN message. If it is not the destination, it simply discards the message. If it is the destination node, node *j* broadcasts the ECN message to its one-hop neighbors. Once a node receives the ECN message from a node in two-hop neighborhood, it sets the HCL flag.

Transmission Rule

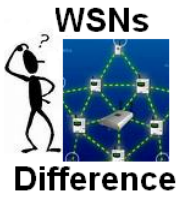
When a node in the network needs to transmit data, it first checks whether it is the owner of the time slot. If the node finds that it is the owner of the time slot then it checks whether the transmission channel is unoccupied. If the node finds the transmission channel is unoccupied, then it transmits the data to the intended destination. Otherwise, it sets a timer and waits for a period of T_0. When the timer expires, it runs the CCA and if the transmission channel is clear, it transmits the data. Otherwise, it waits for random time and repeats the same process. If a node is in high contention level and is the non-owner of the time slot, it postpones the transmission for T_0 time and then performs random backoff within the contention window, $[T_0, T_{n0}]$. When the backoff timer expires the node senses the transmission channel and if the channel is unoccupied then the node transmits the data. Otherwise, the node waits until the channel is clear and repeats the above process.

Receiving Schedule of Z-MAC Protocol

Z-MAC protocol relies on B-MAC [Jpolastre04] protocol for receiving schedule. Z-MAC uses *low power listening* (LPL) mode wherein each node maintains listening duty cycle separated by a check period and each transmission is preceded by a preamble as large as the check period. Therefore, under low duty cycles, the energy consumption of Z-MAC in idle listening is comparable to that of B-MAC. The check period is one of the important factors in the receiving schedule because the check period must allow one complete transmission of the data packet. Thus the size of the slot must be larger than the sum of check period, T_0, T_{n0}, CCA period and time required for propagation of one data packet.

Performance of Z-MAC Protocol

Z-MAC protocol can dynamically adjust the behavior of medium access between CSMA and TDMA depending on the level of contention in the network. The protocol takes advantage of the two-hop neighbors topology information and loosely synchronized clocks to improve MAC performance under high contention. Like TDMA, Z-MAC achieves high channel utilization under high contention and reduces collision among two-hop neighbors at a low cost. Under low contention, the protocol behaves like CSMA and achieves high channel utilization and low latency. A unique feature of Z-MAC is that its performance is robust to synchronization errors, slot assignment failures and time varying channel conditions. In the worst case, its performance always falls back to that of CSMA. Comparing to B-MAC [Jpolastre04], Z-MAC has advantage under medium to high contention and is competitive under low contention (especially in terms of energy efficiency).



Sift MAC [Kjamieson03] shows high performance under one-hop contention, but under two-hop contention, it needs to rely on RTS/CTS and incurring high overhead. Z-MAC can be favorably adopted in applications where expected data rates and two-hop contention are medium to high.

3.7 Conclusions

In this chapter, we have gone through the challenges of MAC design in wireless sensor

networks. To resolve these challenges, much research in the literature has been conducted on design of effective MAC protocols suitable for different WSN applications. Hence, this chapter also briefly introduced several classical MAC protocols including contention-based S-MAC and T-MAC, schedule-based TRAMA as well as hybrid and event based MAC protocols like Sift MAC, Z-MAC and B-MAC protocols.

Problems & Exercises

3.1 Multi-choice questions:

(1) Which one is not a state in TRAMA protocol?

A. Sleep state

B. Receive State

C. Transmit State

D. Wake up state

(2) The size of contention window in Sift MAC protocol is

A. 16

B. 32

C. 512

D. None of the above

(3) Z-MAC protocol combines two traditional protocols for medium access control. Which two?

A. CDMA and TDMA

B. FDMA and CSMA

C. CDMA and TDMA

D. CSMA and TDMA

3.2 Why is energy an important concern in the design of medium access protocols in WSNs?

3.3 Does the performance of Sift MAC protocol depend on the number of the nodes in WSN? What is the reason for the performance change when the number of nodes in the network increases?

3.4 What are the major differences between S-MAC and T-MAC protocol?

3.5 What are the different states for the nodes in WSNs using TRAMA protocol? Explain the operation of different states for nodes in WSNs using TRAMA.

3.6 Explain the operation of each state for nodes using Z-MAC protocol.

3.7 Explain the importance of low power listening and clear channel assignment in B-MAC protocol.

3.8 Explain the early sleeping problem in T-MAC protocol and how to resolve it.

3.9 Explain how the nodes in WSNs using S-MAC protocol choose and exchange their schedules.

3.12 Explain the hidden and the exposed terminal problems in WSNs. Give examples on how to

handle these problems in WSNs.

References

[Abarroso05] A. Barroso, U. Roedig and C. Sreenan, "*μ-MAC: An Energy-Efficient Medium Access Control for Wireless Sensor Networks*", Proceeedings of the 2nd European Workshop on Wireless Sensor Networks, IEEE, pp. 70-80, January, 2005

[Achandra00] A.Chandra V. Gummalla and J.O.Limb, "*Wireless Medium Access Control Protocols*", IEEE surveys and tutorials, Vol. 3, no. 2, Second Quarter, 2000

[Aelhoiydi04] A. El-Hoiydi and J.-D. Decotignie, "*WiseMAC: An Ultra Low Power MAC Protocol for the downlink of Infrastructure Wireless Sensor Networks*", IEEE Computers and Communications, Vol. 1, pp:244-251, July 2004.

[Aelhoiydi05] A. El-Hoiydi and J.-D. Decotignie, "*Low Power MAC Protocol for Infrastructure Wireless Sensor Networks*", Mobile networks and Applications, ACM, Vol. 10, no. 5, pp 675-690, October 2005.

[Awoo01] A. Woo and D.E. Culler, "*A transmission control scheme for media access in sensor networks*", in Proceedings of the 7th annual international conference on Mobile computing and networking, MobiCom '01, pp221-235, July, 2001

[Bwendi02] Wendi. B, Anantha P, Hari Balakrishnan, "*An Application-Specific Protocol Architecture for Wireless Microsensor Networks*", IEEE Transactions on Wireless Communications, vol.1, no. 4, pp660-670, October, 2002

[CcEnz04] Enz.C.C, El-Hoiydi.A, Decotignie.J-D, Peiris.V, "*WiseNET: An ultralow-power wireless sensor network solution*", IEEE Journal, vol. 37, no. 8, pp 62-70, August 2004.

[Eelopez06] E.E. Lopez, J. Vales-Alonso, A. S. Martínez-Sala, J. García-Haro, P. Pavón-Mariño, M.V.B. Delgado, "*A wireless sensor networks MAC protocol for real time applications*",

Personal and Ubiquitous Computing, ACM, Vol. 12, no. 2, pp111-122, January, 2008.

[Fli06] F. Li, Y. Li, W. Zhao, Q. Chen, W. Tang, "*An Adaptive Coordinated MAC Protocol Based on Dynamic Power Management for Wireless Sensor Networks*", in proceedings of the 2006 international conference on Wireless communications and mobile computing, ACM, pp 1073-1078, July, 2006.

[Ftobagi75]F. Tobagi and L. Kleinrock. "*Packet switching in radio channels, part ii: Hidden-terminal problem in carrier sense multiple access and the busy-tone solution*" IEEE Transactions on Communications, vol. 23, no. 12, pp973-977, December 1975.

[Glu04] G. Lu, B. Krishnamachari, C.S. Raghavendra, "*An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks*", in the Proceedings of the 18th International Parallel and Distributed Processing Symposium, IEEE, pp 224-231, April 2004.

[Hschulzrinne96] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "*RTP: A Transport Protocol for Real-Time Applications*", RFC1889, January, 1996

[IEEE07] IEEE Standard for Information technology— Telecommunications and information exchange between systems— Local and Metropolitan area networks— Specific requirements "*Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*", pp120-121, July 2007

[Irhee06] I. Rhee, A. Warrier, J. Min, L. Ki, "*DRAND: Distributed Randomized TDMA Scehduling for Wireless Ad-Hoc Networks*", in the proceeding of MobiHoc, IEEE, pp190-201, May 2006

[Irhee08] I. Rhee, A. Warrier, M. Aia, J. Min, and M.L. Sichitiu, "*Z-MAC: A Hybrid MAC for Wireless Sensor Networks*", IEEE/ACM Transactions on Networking, Vol. 16, no. 3, pp 511-524, June 2008

[Jai04] J. Ai, J. Kong, D. Turgut, "*An adaptive coordinated medium access control for wireless sensor networks*", in proceedings of 9th international symposium on Computer and Communications 2004, IEEE, Vol. 1, pp 214-219, July, 2004.

[Jli04] J. Li, G.Y. Lazarou, "*A bit-map-assisted energy-efficient MAC scheme for wireless sensor networks*", in Proceedings of the 3rd international symposium on Information processing in sensor networks, ACM, pp55-60, April, 2004

[Jpolastre04] J. Polastre, J.Hill and D.Culler "*Versatile Low Power Media Access for Wireless Sensor Networks*", in the proceeding of 2nd international conference on Embedded networked sensor systems, ACM, pp95-107, October, 2004

[Keoliver05] K.E.Oliver "*Introduction to Automatic Design of Wireless Networks*", CrossRoads ACM Student Magazine, Vol.11, no. 4, pp 1-4, 2005

[Kjamieson03] K. Jamieson, H. Balakrishnan, Y.C.Tay, "*Sift: A MAC protocol for Event-driven wireless sensor networks*", Lecture Notes in Computer Science, Springer link, pp 260-275, May, 2003.

[Ksarvakar08] K. Sarvakar, P.S. Patel, "*An Efficient Hybrid MAC Layer Protocol Utilized for Wireless Sensor Networks*", in proceedings of 4th conference on Wireless Communication and Sensor Networks08, IEEE, pp22-26, December, 2008.

[Ksohrabi00]K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie "*Protocols for Self-Organization of a Wireless Sensor Network*", IEEE Personal Communications, Vo. 7, no. 5, pp 16-27, October, 2000

[Lcampelli07] L. Campelli, A. Capone, M. Cesana, E. Ekici, "*A Receiver Oriented MAC Protocol for Wireless Sensor Networks*", in the proceedings of Mobile Adhoc and Sensor Systems 07, pp 1-10, October, 2007

[Ngajaweera08] N. Gajaweera, D. Dias, "*FAMA/TDMA Hybrid MAC for Wireless Sensor Networks*", in proceedings of 4th international conference on Information and automation for sustainability08, IEEE, pp 67-72, December, 2008

[Pkarn90] P. Karn, "*MACA - A New Channel Access Method for Packet Radio*", ARRL/CRRL Amateur Radio 9th Computer Networking Conference, pp 1-5, September, 1990

[Plin04] P. Lin, C. Qiao and X. Wang, "*Medium Access Control with a Dynamic Duty Cycle for Sensor Networks*", in the proceedings of Wireless Communications and networking conference, Vol. 3, pp 1534-1539 ,March, 2004

[Rkannan03] R. Kannan, K. Ram, S.S. Iyengar, V. Kumar, "*Energy and rate based MAC protocol for Wireless Sensor Network*", in the proceedings of ACM SIGMOD 2003, Vol. 32, no. 4, pp 60-65, December, 2003.

[Rramanathan97] S. Ramanathan, "*A unified framework and algorithms for (T/F/C) DMA channel assignment in wireless networks*", in proceedings of IEEE INFOCOM, Vol. 2, pp 900-907, April, 1997

[Sganeriwal03] S.Ganeriwal, R.Kumar, M.B. Srivastava, "*Timing-sync Protocol for Sensor Networks*", in the proceedings of 1st international conference on Embedded networked sensor systems, ACM, pp 138-149, November 2003

[Szhou07] S. Zhou, R. Liu, D. Everitt, J. Zic, "*A^2-MAC: An Application Adaptive Medium Access Control Protocol for Data Collections in Wireless Sensor Networks*", in the proceedings of IEEE ISCIT07, pp 1131-1136, October, 2007

[Tvdam03] T. Van Dam, K.Langendoen, "*An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks*", in proceedings of 1st international conference on Embedded networked sensor systems, ACM, pp171-180, November 2003.

[Vrajendran05] V. Rajendran, J. J. Garcia-Luna-Aceves, K. Obraczka, "*Energy-Efficient, Application-Aware Medium Access for Sensor Networks*", in the proceedings of IEEE Mobile Adhoc and Sensor Systems 05, pp 630-637, November 2005

[Vrajendran06] V.Rajendran, K. Obraczka, J.J. Garcia-Luna-Aceves, "*Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks*", ACM, Vol.12, no. 1, pp 63-78, February, 2006.

[Wstallings04]W. Stallings, "*IEEE 802.11 Wireless LANs: a to n*", IT Pro, Vol. 6, pp 32-37, September-October, 2004

[Wye02] W. Ye, J. Heidemann, D. Estrin, "*An Energy-Efficient MAC Protocol for Wireless Sensor Networks*", in Proceedings of IEEE INFOCOM, Vol. 3, pp 1567-1576, June, 2002

[Wye04] W. Ye, J. Heidemann, D. Estrin, "*Medium Access Control with coordinated Adaptive sleeping for Wireless Sensor Networks*", IEEE/ACM transactions on networking, Vol.12, no.3, pp453-506, July, 2004

[Rappaport96] T.S. Rappaport, *Wireless Communication: Principles and Practices*, Prentice-Hall Upper Saddle River, NJ, 1996

[Bharghavan93] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "*MACAW: A Media Access Protocol for Wireless LAN's*", in the *Proceedings of ACM SIGCOMM Conference* (SIGCOMM '94), August 1993, 212-225

[Stemm97] M. Stemm and R. H Katz, "*Measuring and reducing energy consumption of network interfaces in hand-held devices*," IEICE Transactions on Communications, vol. E80-B, no. 8, pp. 1125–1131, Aug. 1997.

[Bao01] L. Bao and J. J. Garcia-Luna-Aceves, "*A new approach to channel access scheduling for Ad Hoc networks*", in The Seventh Annual International Conference on Mobile Computing

and Networking 2001, pp. 210–221.