Chapter 10
Medium Access and Routing

10.1 Introduction

A wireless sensor network (WSN) consists of a large number of sensor nodes (SNs) that collect data at BS or sink node in a multi-hop fashion. The SNs use a single channel and could have collision present under many different scenarios as the hidden terminal problem was discussed in earlier chapter. Similarly, when SNs send beacon signals to determine neighbours, collision could be present due to asynchronous nature of SNs. SNs sense and provide data from the surrounding area, and the presence of any event is deduced based on the values provided by many SNs.

10.2 Collision Avoidance in a WSN

Important primary attributes of MAC protocols in WSN are collision avoidance, energy efficiency, and scalability in terms of SNs density. The secondary attributes are latency, fairness, throughput, bandwidth utilization, always sensing versus transceiver sleep–awake cycle, and data aggregation. Each SN has limited sensing range of r_s and communication range of r_c. These allow SN to cover a given area and transmit data to BS in multi-hop fashion. So, the sensing range allows SN to determine neighboring SNs. But, SNs far apart do not know the presence of each other, and hidden terminal problem was discussed earlier. This can be avoided by using small hand-shaking packets of RTS-CTS.

When RTS (request-to-send) packet is sent by SN A to SN B as shown in Fig. 10.1, the signal is received by all SNs in the communication range of SN A. RTS also contains the length of packet to B, and all devices in communication range of A knows how long the medium will be kept busy by A and the time period is known as NAV (network allocation vector). If device B is ready to receive, it sends CTS (clear-to-send) packet back to SN A. This message is heard by all

D.P. Agrawal, *Embedded Sensor Systems*,
DOI 10.1007/978-981-10-3038-3_10

devices in communication range of SN B, including the period of follow-up data from A to B. This NAV allows SN C to understand ongoing communication and thus learns the presence of SNs B and A. In a similar way, using TRC packet by SN B and CTS by SN A could avoid exposed terminal problem as shown in Fig. 10.1b. All devices within SN B's RTS packet and handshaking CTS packet by A allow all devices to keep quiet for NAV period. SN C knows the ongoing transmission from B to A and will be able to send data to SN D, thereby avoiding exposed terminal problem (Fig. 10.2).

SNs are distributed randomly in a WSN, and simplest scheme for collision avoidance is Aloha where SN ready to transmit packet immediately as shown in Fig. 10.3a. As packets are of different size and slots are not synchronized, collision could occur either at the beginning of a transmitted packet or at the end of the packet. An improvement occurs when slot size is fixed and slots are synchronized. Throughput is increased in slotted Aloha protocol and is illustrated in Fig. 10.3b.

In a WSN, packets are transmitted as fixed size packets, and time is basically slotted. As many distributed sender SNs may want to communicate at the same time, the question is how to avoid or minimize collision. TDMA schedule can be used if WSN is partitioned into clusters, and CH of each cluster dictates the TDMA schedule. The other option before forming clusters is to use contention-based scheme. CSMA/CA (carrier sense multiple access/carrier avoidance) shown in Fig. 10.2a is somewhat similar to IEEE 802.11 ad hoc mode CSMA/CA where SN senses the medium if some other SN is using; if so, then wait till it becomes free and then wait for random period between 1 and CW-1 (Contention window-1). A collision between transmissions from two or more SNs is possible and is indicated by the absence of ACK message from destination SNs if the same random delay is selected by transmitting SNs. In such a situation, the CW is doubled and the same process is repeated with the hope that colliding SNs will generate different random delays with increased window size. Collision is known only to colliding SNs as they do not receive ACK message, and the remaining SNs will keep the initial value of CW. The whole process is known as CSMA/CA.

In CSMA/CA, after sensing the medium, the SN can start transmitting immediately after the delay counter becomes zero and that is called persistent CSMA/CA protocol. If the SN waits for one more slot with probability $(1 - p)$ before starting the transmission of a packet, it is called non-persistent CSMA/CA (Fig. 10.2b).

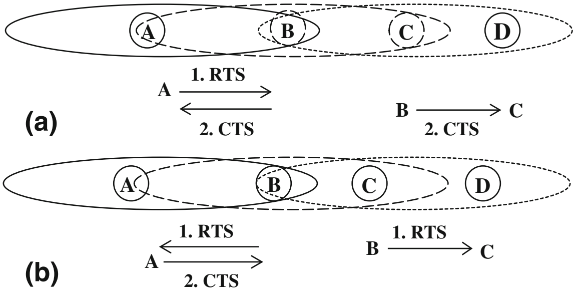


Fig. 10.1 **a** Avoiding hidden terminal problem using RTS/CTS, **b** avoiding exposed terminal problem using RTS/CTS

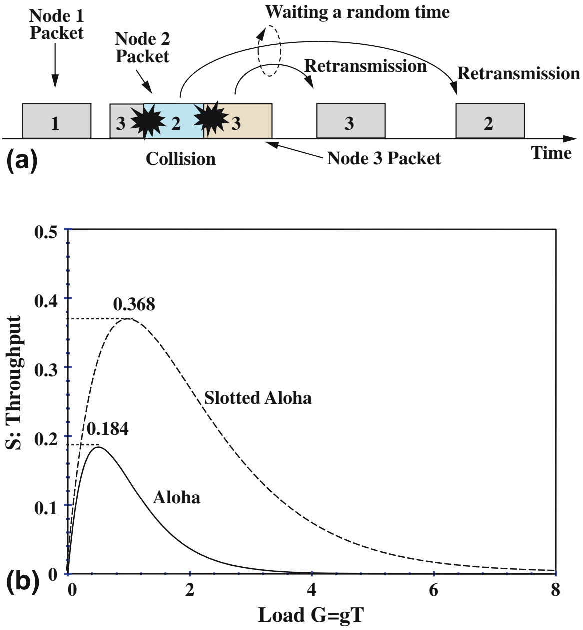


Fig. 10.2 **a** Collision in Aloha protocol, **b** throughput of Aloha/slotted Aloha protocols

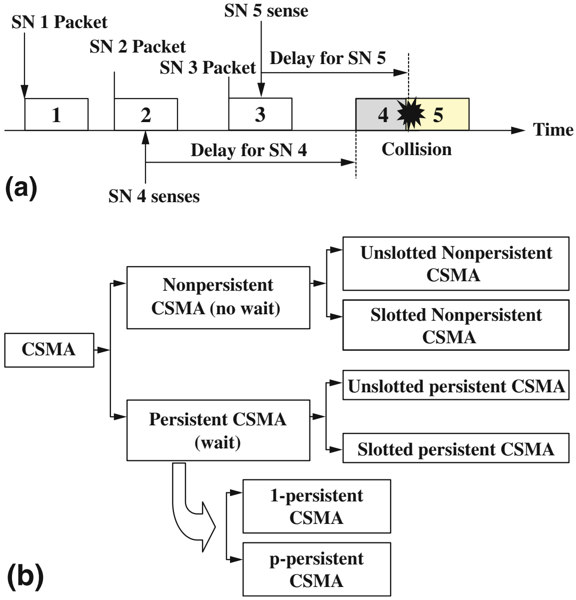


Fig. 10.3 **a** Collision in CSMA/CA, **b** classification of CSMA/CA schemes

Here, *p* indicates the probability of persistent wait for a slot by a SN. The use of random delay is shown in Fig. 10.4a. The counter for second device is frozen as soon as the first SN starts transmitting. In addition, DIFS (distributed interframe space) is also added to ensure better functioning of CSMA/CA as shown in Fig. 10.4b. In order to completely avoid collision, similar to ad hoc networks, RTS/CTS-based handshaking mechanism can be used (Fig. 10.4c) between a sender and receiving SNs before transmitting actual data as summarized in Table 10.1. A good trade-off remains between non-persistent and 1-persistent CSMA, and throughputs are compared with basic Aloha and slotted system in Fig. 10.5.

The SNs provide their location as well as event number. Thus, ID of SNs is not very critical and what is important is the values provided and WSN is considered as "data-centric" as it may require a huge ID due to large number of deployed SNs. In addition to location, SNs provide associated attribute values. So, unlike ad hoc networks, SN-to-SN routing is not required and data need to be routed from SNs to BS or sink node. Once SNs are deployed randomly, each SNs determine their adjacent neighbors by sendingbeacon signals. Then, the SNs could be grouped together to form clusters, and all SNs of a cluster could elect a cluster head (CH) that would be responsible to communicate with the BS/sink node. SNs measure physical parameter from the surrounding area; the reading provided by

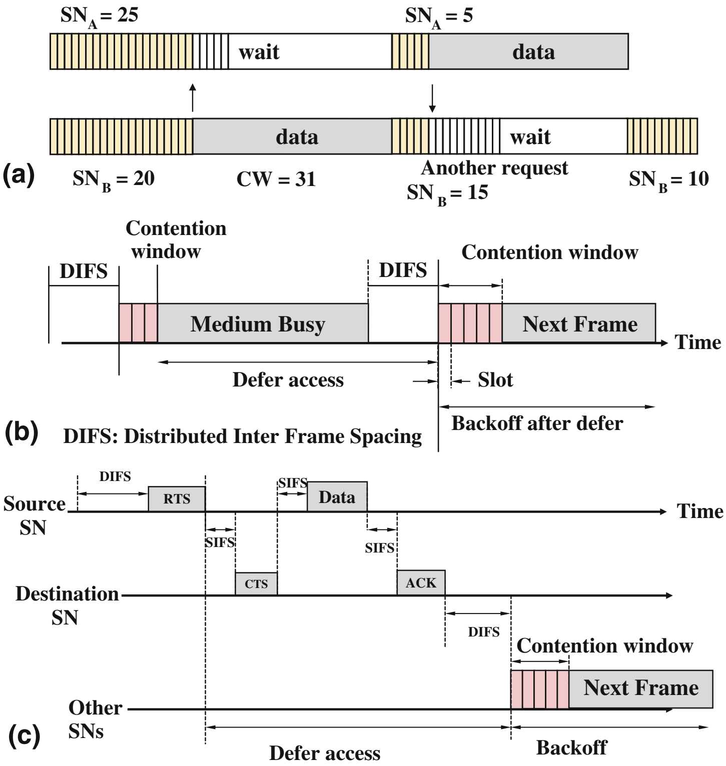


Fig. 10.4 **a** CSMA/CA illustrated, **b** CSMA/CA with DIFS, **c** CSMA/CA with DIFS and RTS/CTS

Table 10.1 Steps for *p*-persistent CSMA protocol

Step 1: If the medium is idle, transmit with probability p, and delay for worst case propagation delay by one packet with probability $(1 - p)$
Step 2: If the medium is busy, continue to listen until medium becomes idle, then go to Step 1
Step 3: If transmission is delayed by one time slot, continue with Step 1 $p = 0$: non-persistent and $p = 1$, 1-persistent CSMA

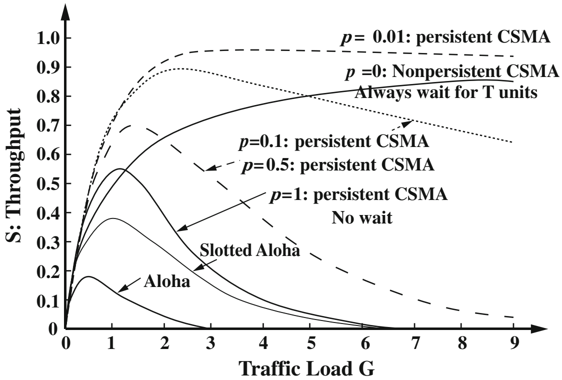


Fig. 10.5 Throughput comparison between persistent and non-persistent CSMA/CA

adjacent SNs could be similar or close to each other, and it is desirable to combine such values together, commonly known as data aggregation. This process can be done at each CH as it can collect data from cluster member SNs, and in this way, the energy dissipation through WSN is also controlled and is discussed later.

10.3 Routing in a WSN

In a WSN, data need to be sent from SNs to BS/sink node in multi-hop fashion. BS/sink node needs to broadcast a query to all SNs, indicating which physical parameters need to be forwarded and what frequency they need to be sent. You may have CHs if a WSN is partitioned into multiple clusters and each cluster elects a CH (Cluster Head). Then, the packet could travel to CH from associated SNs and then CH can combine (aggregate) data to the BS. In brief, routing is needed in a WSN, and many results obtained for ad hoc networks can be used effectively in a WSN. As discussed earlier, beacon signals are used to determine neighbors and 2-hop neighbors can be obtained by piggybacking 1-hop neighbor information. In the previous section, we discussed how to utilize CSMA/CA effectively to schedule transmission from SNs with sleep–awake cycle.

In a WSN, two different routings are needed: one dissemination of query information from the BS/sink node to all SNs, and routing of sensed data from responding SN to the BS. BS could broadcast the query to all SNs in a single high power transmission as shown in Fig. 10.6. In another scheme, BS is assumed to be far away and query from BS cannot reach all SNs in a single transmission. Hence, query in directed diffusion protocol [1] has to be diffused throughout WSN in multi-hop fashion between SNs (Fig. 10.7a). While the query is diffused from BS/sink node of the WSN, a primary path is selected from each SN to BS that requires minimum number of hops and is termed as gradient determination. If there are multiple paths with the same hop-length from a SN to BS, then other criterion such as packet delivery ratio can be used to select among them as detailed in Fig. 10.7b. It is interesting to note that an event could start from some specific points and move outwards to reach the requesting SN. So, a small number of SNs can be reinforced to prevent further flooding in the WSN.

Three types of queries are possible and are either historical queries with analysis of data collected over time, one-time queries giving snapshot view of the network, or persistent queries that involves periodic monitoring of data at regular intervals for a long period of time. Historical query is mainly used for the analysis of historical data stored at the BS, e.g., “What was the temperature 2 h back in the northwest quadrant?” One-time query gives a snapshot view of the network, e.g., “What is the temperature in the northwest quadrant?” Persistent query is mainly used to monitor a network over a time interval with respect to some physical parameters, e.g., “Report the temperature in the northwest quadrant for the next 2 h.”

The routing required to respond to a query is application-specific and data-centric, while data aggregation capability with clustered WSN is desirable so as to minimize energy consumption. The query could be to respond either proactive or reactive way. In proactive approach, the periodic-sensed value is sent by SN, e.g., every 2 min for 20 h. In reactive scheme, sensed value is sent if it crosses a threshold, e.g., whenever temperature is beyond 35 °C. Hybrid combines both proactive and reactive approaches, e.g., reactive and proactive with increased reporting time (e.g., every 2 h). In proactive protocol, the SNs in a WSN periodically switch on their SN’s transceiver, *sense the environment, and record the data of interest and transmit it to the during* allocated time slot by the CH (Fig. 10.8a). WSNs in responding to a query could follow either flat or clustered topology architecture and classification of different routing protocols is summarized in

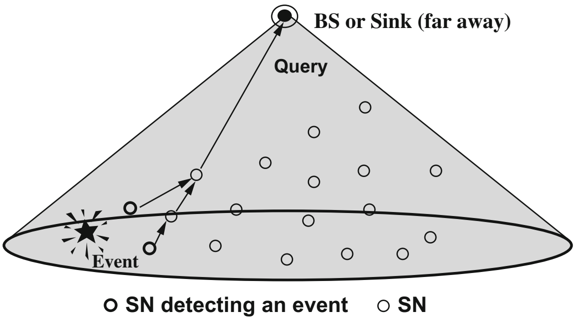


Fig. 10.6 Broadcasting of a query from BS/sink to SNs in a WSN

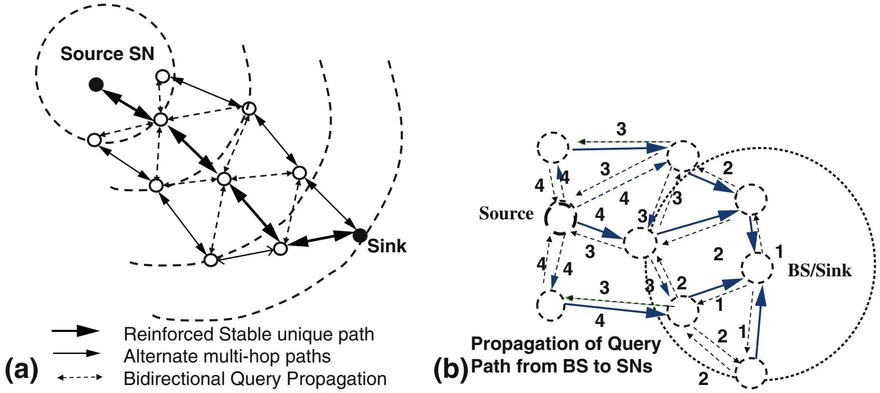
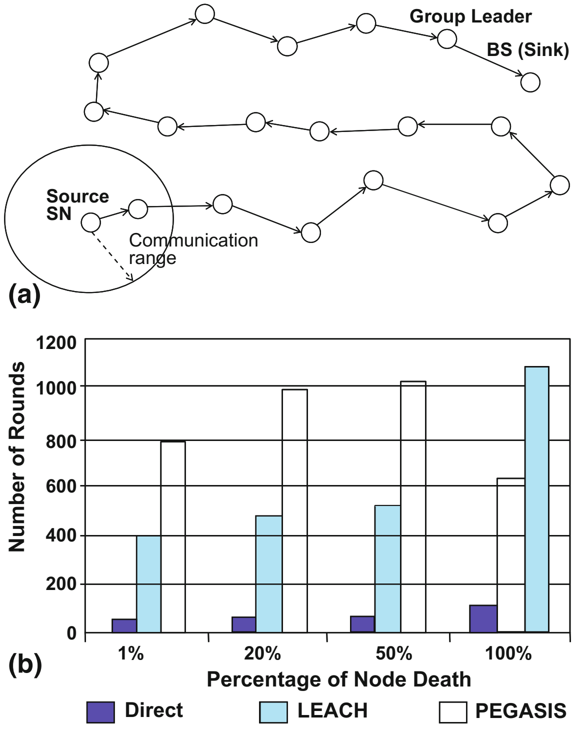

Source SN
Sink
(a)
Reinforced Stable unique path
Alternate multi-hop paths
Bidirectional Query Propagation
Source
BS/Sink
1
2
3
4
Propagation of Query
Path from BS to SNs
(b)

Fig. 10.7 **a** Directed diffusion of query from BS/sink node to SNs, **b** propagation of query in directed diffusion from BS/sink node to SNs

Table 10.2. Since the SNs switch off their transceiver at all times except the report times, the energy of the WSN is conserved. Report time T_R is the time period between successive reports sent by a transceiver of the SN and attributes (A) is a set of physical parameters which the BS is interested in obtaining data about. It is possible that the time-critical data may reach the BS only after report time.

Fig. 10.8 **a** PEGASIS scheme, **b** comparing energy consumption of PEGASIS with direct scheme



As more energy is consumed by SN in transmitting and receiving data and SNs near BS face the problem of energy-hole problem, efforts ought to be made in making the number of transmissions/reception per SN equal. To achieve equalization of energy consumption in a WSN, a novel scheme known as PEGASIS [2] has been introduced where in a continuous traversal between SNs and BS is determined as shown in Fig. 10.8a. By having a single path, all SNs have to transmit one packet and receive one packet (except the source SN at one end transmits a packet and BS at the other end receives a packet). This balances energy consumption in a WSN and performs much better than direct transmission to BS (Fig. 10.8b).

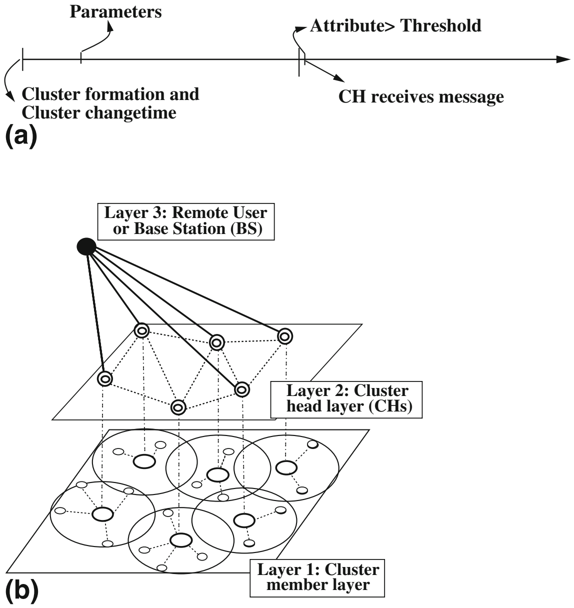
In principle, clustering has been suggested to limit the number of transmissions to BS and conserve energy. Minimum transmission energy (MTE) scheme selects path [3] between SNs and BS so as to minimize energy consumption in transmission and reception by the transceivers. In LEACH (low energy adaptive clustering hierarchy) protocol [3], clusters are formed and CH is selected and changed dynamically based on residual energy of SNs. The advantage is that SNs within a cluster and from neighboring area need to communicate only with the CH which can aggregate received similar data and send a single packet to the BS/sink node. LEACH works in rounds, each with short setup and long steady state. In this proactive protocol, the entire WSN is divided into a number of clusters; all SNs transmit only to their immediate CH (Fig. 10.9b) and CHs at increasing levels in the hierarchy need to transmit data over relatively longer distances. The set-up phase is subdivided into self-advertising as a CH, SNs opting for a cluster, and TDMA schedule assigned by the CH to each SN. In steady state, TDMA is used for data transmission as everyone uses the same channel. CHs use different CDMA codes to communicate with BS/sink node and the hierarchy can be extended to hierarchical clustering. The algorithm is totally distributed, and any SN having larger residual energy can be CH.

LEACH, being proactive, transmits data periodically and any event detected by a SN has to wait for allocated time slot and in many applications that may not be acceptable. TEEN (threshold-sensitive energy-efficient sensor network) has been introduced [4] as a reactive protocol which uses LEACH-based clustering and SNs dynamic reconfiguration capability that suits for time-critical applications. In TEEN , a SN transmits only if sensed value is greater than a hard threshold (H_T),

Table 10.2 Classification of different routing protocols

Broadcast	Unicast			Multi-cast
	Proactive	Reactive	Hybrid	
– Direct	– Pegagus – Leach	–TEEN – Minimum cost forwarding – Rumor routing – Random walk	– APTEEN – Two-tier data dissemination – Directed diffusion	– SPIN

Fig. 10.9 **a** Periodic cluster formation, **b** clustering of SNs in a WSN

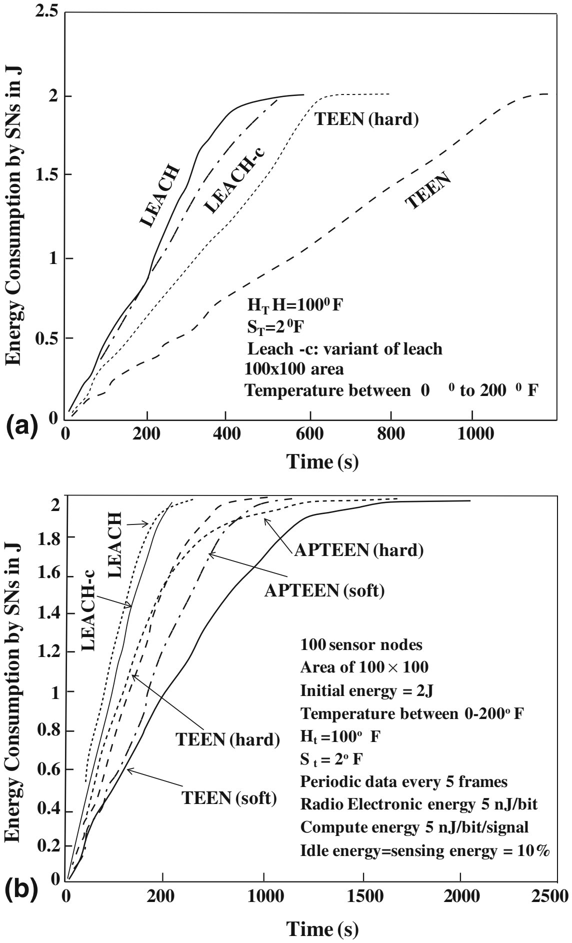

Parameters
Attribute> Threshold
Cluster formation and Cluster changetime
CH receives message
(a)
Layer 3: Remote User or Base Station (BS)
Layer 2: Cluster head layer (CHs)
Layer 1: Cluster member layer
(b)

and it transmits data if the sensed value differs from last transmitted value by more than a soft threshold (S_T). An important feature of this scheme is that time-critical data reaches the BS almost instantaneously. As message transmission consumes much more energy than data sensing, energy consumption can be controlled by changing the threshold values and it offers flexibility by allowing the user to set the threshold values for the attributes (Fig. 10.10a). Attributes can be changed as every cluster change. The only drawback of this scheme is that if threshold is not reached, then the user never gets to know about the SNs. But, it is useful for time-critical environment-like intrusion detection, etc.

If the threshold is never crossed, TEEN will not provide any value and one would start wondering whether SNs are working correctly or not. To address this shortcoming, APTEEN (adaptive periodic threshold-sensitive energy-efficient sensor network protocol) [5] was introduced that combines features of reactive and proactive networks by having reactive element as it is and drastically increasing the sampling period for proactive component that could indicate health of SNs. It offers a lot of flexibility by allowing the user to set the time interval and the threshold values for the attributes. Similar to TEEN, the energy consumption can be controlled by changing periodic interval as well as the threshold values, and the hybrid network can emulate a proactive network or a reactive network, based on the application, by suitably setting the periodic interval and the threshold values as it combines both proactive and reactive policies (Fig. 10.10b).

Sensor protocol for information via negotiation (SPIN) [6] protocol allows network-wide broadcast by limiting negotiation and using local communication. Blind flooding causes excessive resources consumption, and this problem is limited

Fig. 10.10 **a** Energy consumption by TEEN, **b** comparison of energy consumed by APTEEN



by getting localized information and detecting overlapping regions as the same data available from many neighbors. Broadcast is limited in a WSN with *n*SNs, and every SN handles O(*n*) messages. Two versions of protocols SPIN-PP (point to point) and SPIN-BC (local broadcast communication) [7, 8] require data to be updated throughout WSN. In SPIN-PP, SN with data advertises (ADV) to all its neighbors (Fig. 10.11a) and receiving SNs requests for data. The receiving SN follows the same sequence by advertising the presence of data to its neighbors. This process continues around the network, and SNs may aggregate their data to ADV. In a lossy network, ADV may be repeated periodically and REQ if not answered. SPIN-BC is a variant that supports broadcast operation in a WSN so as to reach all neighbors.

Random walk scheme depends on finding a random walk over a grid with little state information, and SNs are assumed to be located at cubic grid junctions. With

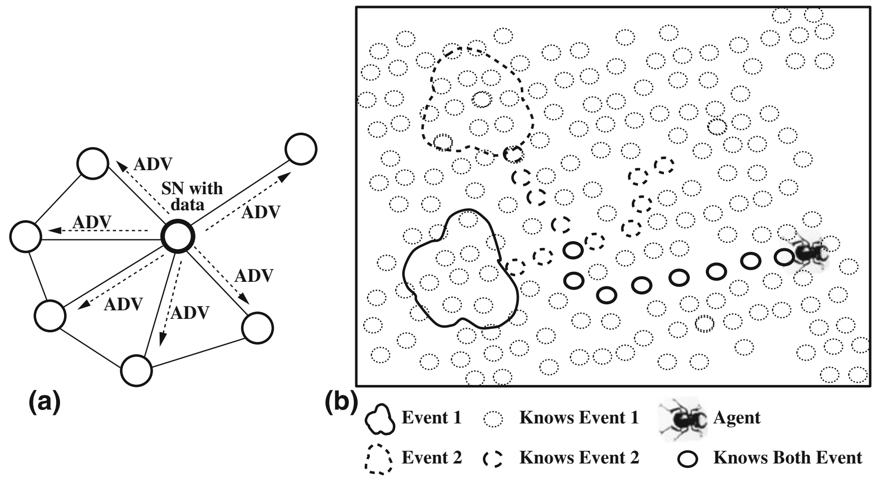


Fig. 10.11 **a** ADV message sent to neighbors by SN having data in SPIN-PP, **b** rumor routing in a WSN

localized communication, coordinates differences (D_x, D_y) using Distributed Bellman Ford using local communication. Every SN computes the probability of moving on X and Y directions and in each move to an adjacent SN on X or Y using computed probability. By carefully adjusting the probability, load balancing can be achieved by selecting diverse routes at different time. If one adjacent SN is missing, then go to the other SN.

If both are missing, then go to a neighbor whose breadth-first distance to the destination is strictly smaller than the current node. In rumor routing (Fig. 10.11b), the network follows a dynamic graph [9] as SNs may sleep and wake. For many applications, any arbitrary path will be able to do the job and there is no need for the shortest path as SNs are densely distributed. The scheme is specifically attractive when the ratio of events with queries is within a threshold where it is not attractive to flood the network. Optimal parameters heavily depend on topology as it does not guarantee delivery. The movement on the network is done by several agents, randomly trying to walk straight as every SN maintains a list of neighbors and events. An agent coming from an event is updating SNs it visits.

10.4 MAC Challenges for a WSN

Traditionally, there are many challenges in a WSN at the medium access control level, including fairness, latency, and throughput. For a WSN, both power efficiency and scalability are important as WSNs are designed to operate unattended for long time as it is rather impractical to replenish the batteries. However, SNs are in idle state for most time when no transmission of sensed data occurs.

Measurements have shown that a typical radio transmitter of a SN consumes a similar level of energy in idle mode as in receiving mode. Therefore, it is important that SNs are able to operate in low duty cycles as idle listening consumes significant energy and it is better to have periodic listening and turn off the radio when sleeping most of the time. For example, duty cycle can be reduced to ~10% (200 ms on and 2 s off). Sleep/awake cycle can be used in a randomly deployed WSN and only one set is active at a time among 3 sets (Fig. 10.12a). This is done by organizing available SNs into mutually exclusive sets and they periodically listen and sleep rest of the time. Both *transmitter and receiver need to be awake simultaneously to have data transfer. It is preferable for the* neighboring SNs to have the same schedule for easy broadcast and maintaining low control overhead.

As both *transmitter and receiver SNs need to be awake simultaneously to have data transfer, it is better if* neighboring SNs have the same schedule. Therefore, it is not only required to synchronize SNs but also desirable to remember neighbors' schedule to help when to send data to them (Fig. 10.12b). Each SN broadcasts its schedule few periods of its sleeping and listening cycle so that adjacent SNs are resynchronized with receiving schedule update. The scheduled packets also serve as beacons for new SNs to join neighborhood. A SN wakes up, indicating high signal level while the channel is free and it simply wastes energy contributing as false positive. SN fails to wake up indicating low signal level while the channel is busy and could miss important data and is termed as negative false wakeup. An important problem is how RSSI (receive signal strength indicator) varies and how can RSSI be used to detect presence of a signal. The real question is what is the right wake-up threshold value. In a WSN, BS sends command and collects data while SN side can send or receive tone, and very little work has been done on physical layer protocols for WSNs. Most of the existing MAC works make SNs sleep as long as possible and include at least some aspects of TDMA. The channel could be allocated either as static or dynamic. In static channel allocation with N SNs, available bandwidth is divided into N equal portions and used in FDMA,

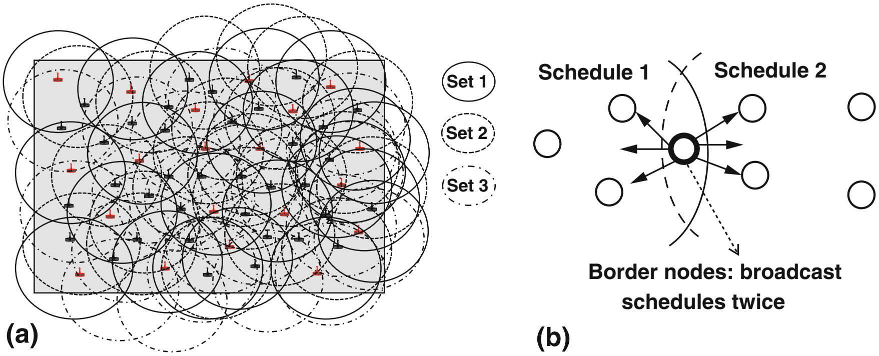


Fig. 10.12 **a** Three set of SNs covering the area for sleep–awake cycle in a WSN, **b** sleep–awake cycle between 2 groups in a WSN

TDMA, CDMA, SDMA or OFDM or ultra-wide band. In dynamic channel allocation, there is no fixed assignment of bandwidth as the number of active SNs changes dynamically, and contention-based allocation scheme is followed. In a WSN with clusters, the number of SNs per cluster is kept small, encouraging the use of static channel allocation. TDMA is suitable for either proactive or reactive type of WSN. In reactive networks, a sudden change could be noticed immediately by many SNs and that could possibly lead to collisions, and it is better to employ TDMA. The use of CDMA by CHs avoids collisions due to the use of the same channel by different clusters and use of TDMA/CDMA avoids intra/inter cluster collisions. When SN fails, MAC and routing protocols must accommodate the formation of new links and routes to other SNs and the BS. The traditional MAC protocols have negative impact in terms of collisions, overhearing, control packets overhead, and idle cycle time. The issue is if an entire message is process in-network in a WSN, it requires *entire* message. So it is advisable not to interleave different messages, and rather a long message is fragmented and sent in burst as RTS/CTS mechanism can reserve medium for entire message and fragment-level error recovery is possible by extending transmitting time and re-transmit immediately if error. Other SNs sleep for whole message time. Overhearing can be avoided when SNs r*eceive packets destined to other SNs. This can be done if SNs sleep when neighbors talk.* So, all immediate neighbors of sender and receiver SNs should sleep *as the* interval *informed by* each packet to other SNs to remain in the sleep mode.

10.5 S-MAC Protocols with Sleep–Awake Cycles

SNs periodically sleep in a WSN and energy efficiency is traded for lower throughput and higher latency as SN has to sleep during other SNs transmissions. The sensor-MAC (S-MAC) [9] protocol explores design trade-offs for energy conservation in the MAC layer. It reduces the radio energy consumption from

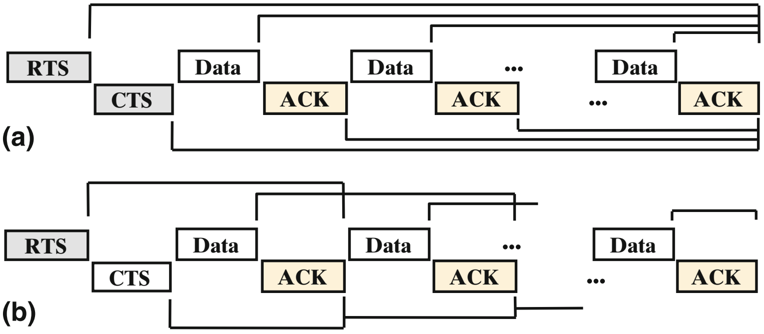
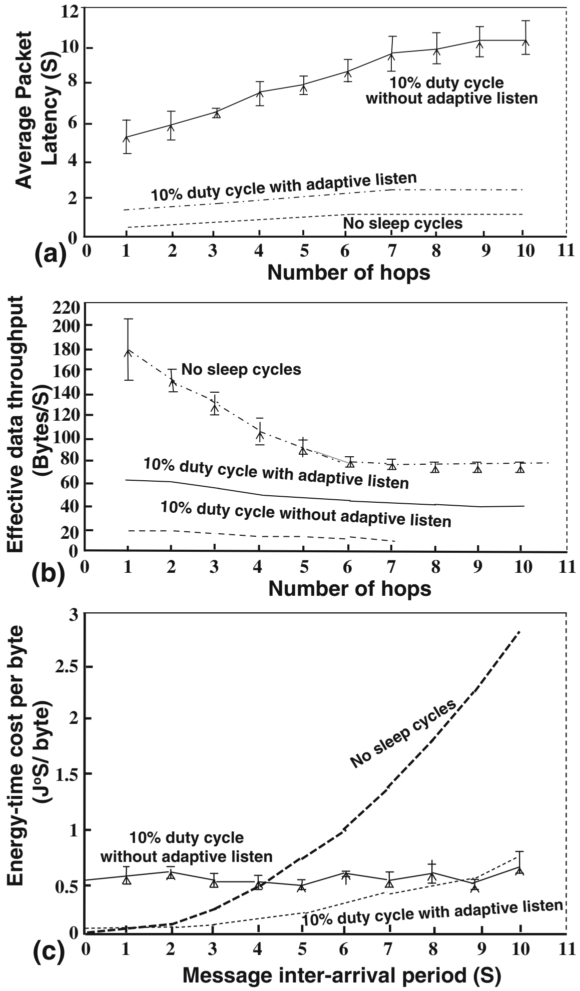


Fig. 10.13 **a** S-MAC message passing, **b** fragmentation in fixed data size

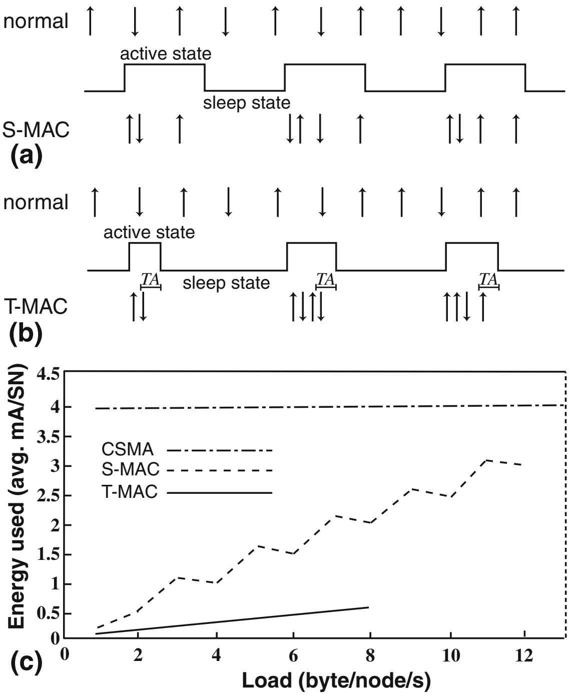
Fig. 10.14 **a** Average packet latency in S-MAC, **b** throughput in S-MAC, **c** energy savings by S-MAC



collision, control overhead, overhearing unnecessary traffic, and idle listening. The basic scheme of S-MAC is to put all SNs into a low-duty-cycle mode of listen and sleep periodically. When SNs are listening, they follow a contention rule to access the medium, which is similar to the IEEE 802.11 DCF. The major components in S-MAC are periodic listen and sleep, collision avoidance, overhearing avoidance, message passing, and synchronize every 10 s. In S-MAC, SNs exchange and coordinate on their sleep schedules rather than randomly sleep on their own and before each SN starts the periodic sleep, it needs to choose a schedule and broadcast it to its neighbors.

To prevent long-term clock drift, each SN periodically broadcasts its schedule as the SYNC packet. To reduce control overhead and to simplify broadcasting, S-MAC encourages neighboring SNs to choose the same schedule, but it is not a

Fig. 10.15 **a** S-MAC, **b** T-MAC, **c** energy saving in T-MAC



requirement. A SN first listens for a fixed amount of time, which is at least the period for sending a SYNC packet. If it receives a SYNC packet from any neighbor, it will follow that schedule by setting its own schedule to be the same (Fig. 10.13a). Otherwise, SN will select an independent schedule after the initial listening period. Such listening period considerably longer than the clock drift. The neighboring SNs exchange SYNC messages while exchanged timestamps are relative rather than absolute. Utilization of RTS/CTS avoids hidden terminal and contains expected duration of message (Fig. 10.13b). Every packet is acknowledged, and adaptive listening can be used so that potential next hop SNs wake up in time for possible transmissions. Performance of S-MAC is given in Fig. 10.14.

T-MAC (timeout MAC) is a RTS/CTS/ACK-based scheme and transmits all messages in bursts of variable length and sleep between bursts. The synchronization is similar to S-MAC (Fig. 10.15a, b). T-MAC is observed by simulation to save energy compared to S-MAC (Fig. 10.15c), while the “early sleeping problem” limits the maximum throughput.

The objective behind B-MAC (Berkeley MAC) [10] is to have low power operation, effective collision avoidance, simple implementation (small code), to be efficient at both low and high data rates, reconfigurable by upper layers, tolerant to changes in the network, and scalable to a large number of nodes. In B-MAC, RTS/CTS, ACKs, etc. are considered higher layer functionality and low power listening (LPL) using preamble sampling (periodic channel sampling) and hidden

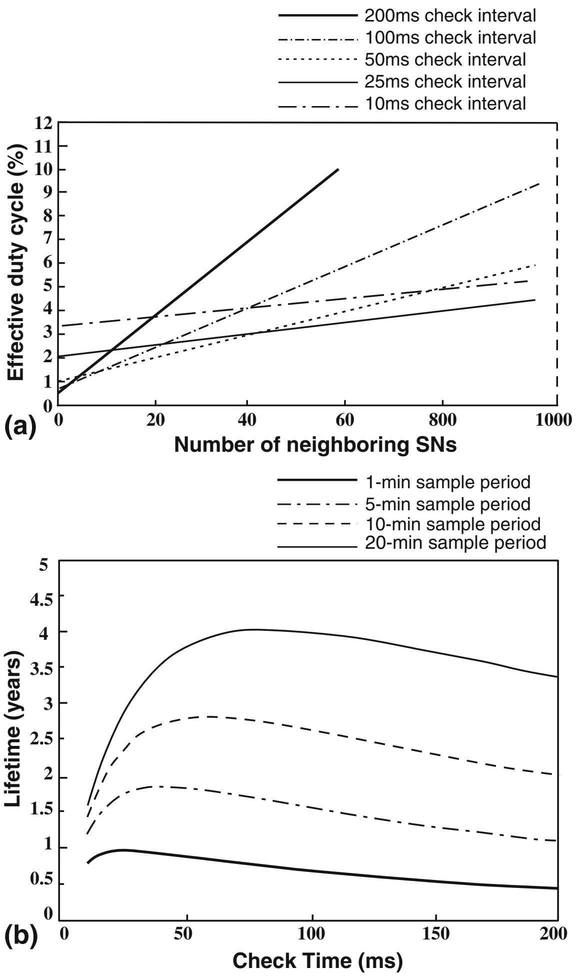


Fig. 10.16 **a** Duty cycle of B-MAC. **b** WSN lifetime in years [10]

terminal and multi-packet mechanisms not provided, should be implemented, and can be implemented by higher layers if needed. Initial and congestion back-off is considered on a per packet basis. Clear Channel Assessment (CCA) could be on or off. The protocol establishes noise floor and then a transmitting SN starts monitoring RSSI of the channel. If a sample has considerably lower energy than the noise floor during the sampling period, then the channel is free and no one is using it. The goal is to minimize listen cost (Fig. 10.16a). The SN periodically wakes up, turns radio on, and checks channel—if energy is detected, node powers up in order to receive the packet. Then, the SN goes back to sleep. If a packet is received after a timeout, then the preamble length matches channel checking period. No explicit synchronization is needed. The energy consumption can be minimized by varying check time/preamble if constants, sample rate, and neighboring nodes are known

(Fig. 10.16b). B-MAC performs better than the other studied protocols in most cases.

P-MAC (Pattern MAC) [11] is based on scheduling received slots as every SN announces its sleeping pattern for the next frame; n-sleep/1 awake cycle. The transmission time is equal to contention window + RTS + CTS + DATA + ACK time and simulated results are observed to be better than SMAC. This protocol is good for relatively stable traffic conditions as adaptation to changes in traffic might be slow. Loose synchronization of time is required.

10.6 Conclusions

Self-organization is a must in any WSN. Address of a SN is not critical as compared to the location of SN where an event has occurred. Routing from SNs to BS is needed, and load balancing in energy consumption of a WSN is important as it is desirable to maintain the same residual energy among SNs. Sleep cycle is useful for enhancing lifetime of a WSN. Collision avoidance is desirable for Beacon signals as well as simultaneous transfer of data while data aggregation could prove to be very useful. How to merge heterogeneous data is still an open question.

10.7 Questions

Q.10.1. What is the goal of localization in a WSN?

Q.10.2. Why do you need SNs to self-organize in a WSN?

Q.10.3. What do you do in self-organization?

Q.10.4. Three beacons are located at $a = (1, 1)$, $b = (1, -1)$, and $c = (-1, 1)$. The received power from nodes a, b, and c is 1.2, 1.5, and 1.7, respectively. Calculate the unknown position of the receiver through a weighted centroid computation.

Q.10.5. What is meant by 1-hop and 2-hops neighbors?

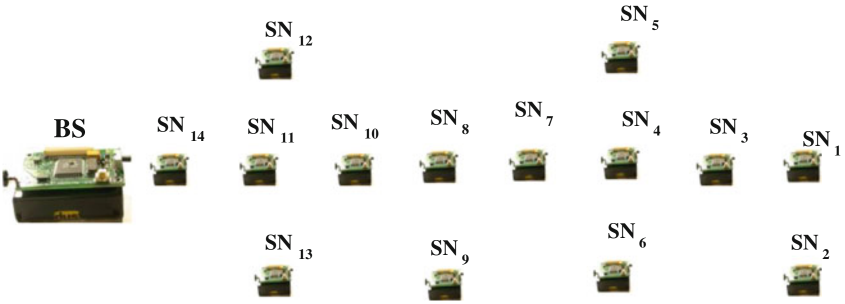
Q.10.6. What is the use of piggybacking 1-hop neighbor information?

Q.10.7. Why do you need routing from SN to BS?

Q.10.8. Do you need routing of data from SN to another SN? Justify your answer.

Q.10.9. What is the role of sleep–awake cycle in life-time of a WSN?

Q.10.10. A WSN is given in the following figure; SN_1 needs to send data to the BS. In the second of routing, a path can be selected either passing through SN_4, SN_5, or SN_6. Under what condition, a particular path will be followed?



Q.10.11. Explain the following terms:

(a) Sleep–awake cycle
(b) Data aggregation
(c) One-hop neighbors
(d) Anchor nodes
(e) Directed diffusion
(f) Synchronization
(g) Intrusion detection using WSN

References

1. Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," www.isi.edu/div7/publication_files/directed_diffusion_scalable.pdf.
2. Stephanie Lindsey and Cauligi S. Raghavendra, "PEGASIS: Power-Efficient Gathering in Sensor Information Systems," http://ceng.usc.edu/~raghu/pegasisrev.pdf.
3. Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," Proceedings of the Hawaii International Conference on System Sciences, Maui, Hawaii, January 4–7, 2000.
4. Arati Manjeshwar and Dharma P. Agrawal, "TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks," Parallel and Distributed Processing Symposium, International. Vol. 3. IEEE Computer Society, 2001.
5. Arati Manjeshwar, Qing-An Zeng, and Dharma P. Agrawal, "An Analytical Model for Information Retrieval in Wireless Sensor Networks Using Enhanced APTEEN Protocol," IEEE Transactions on Parallel and Distributed Systems, vol. 13, no. 12, pp. 1290–1302, DECEMBER 2002.
6. Jaewan Seo, Moonseong Kim, Sang-Hun Cho, and Hyunseung Choo, "An Energy and Distance Aware Data Dissemination Protocol Based on SPIN in Wireless Sensor Networks," ICCSA 2008, Part I, LNCS 5072, pp. 928–937, 2008.
7. Luis Javier García Villalba, Ana Lucila Sandoval Orozco, Alicia Triviño Cabrera and Cláudia Jacy Barenco Abbas, "Routing Protocols in Wireless Sensor Networks," *Sensors*, vol. *9, no.* 11, pp. 8399–8421, 2009.

8. Sergio D. Servetto and Guillermo Barrenechea, “Constrained Random Walks on Random Graphs: Routing Algorithms for Large Scale Wireless Sensor Networks,” Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, WSNA’02, pp. 12–21, September 28, 2002.
9. Wei Ye, John Heidemann, Deborah Estrin, “An Energy-Efficient MAC Protocol for Wireless Sensor Networks,” INFOCOM 2002.
10. Joseph Polastre, Jason Hill, and David Culler, **“**Versatile low power media access for wireless sensor networks,” Proceedings of the 2nd international conference on Embedded networked sensor systems, pp. 95–107, SenSys’04.
11. Tao Zheng, S. Radhakrishnan, and V. Sarangan, “PMAC: an adaptive energy-efficient MAC protocol for wireless sensor networks,” Proceedings 19th IEEE International Parallel and Distributed Processing Symposium, 4–8 April, 2005.