# Chapter 5
# Clock Synchronization and Localization

## 5.1 Introduction

WSNs have numerous applications and are usually embedded in the environment. The basic objective in a WSN is to collect sensed data from a large number of SNs at one location commonly known as BS (or sink node), analyze the data to monitor the environment, and eventually control the actuator. This is to be done for a long period of time without any human intervention. The SNs are usually miniature, lightweight, inexpensive, and energy-efficient and need to keep transceiver in sleep mode as long as possible in order to conserve energy. On the other hand, when data are sent from one SN to another SN on way to BS, both SNs need to be awake. So, there is a need for very tight synchronization between SNs. In that respect, any SN's clock can be used as a reference. But from overall access and logistic point of view, it is rather better to consider clock of a BS as the reference node. BS sends a query to all SNs, indicating which physical parameters are desirable and at what frequency rate. BS usually transmits that in one transmission at higher power level such that all SNs receive that requested query in a single broadcast.

Each SN has a sensing range $r_s$ and the communication transmission range $r_c$ (Fig. 5.1a), which determines distance to SN's neighbors. Such a WSN can be represented by a network graph as shown in Fig. 5.1b. The communication is assumed to be symmetric. This means if SN A can communicate with SN B, then SN B can also communicate with SN A and undirected graph simply indicates that.

If two neighboring SNs transmit data at the same time, collision occurs and received data cannot be interpreted correctly (Fig. 5.2a). That means, all collided packets need to be retransmitted. In Fig. 5.2b, adjacent SNs a and b know the presence of each other and so do SNs b and c. But, a does not know the presence of c, and if both a and c transmit data to b at the same time, collision occurs. This is known as hidden terminal problem, and a is termed as hidden from c and vice versa. Figure 5.2c illustrates exposed terminal problem where SN c cannot communicate with SN d as SN b is having ongoing communication with SN a and detecting and

D.P. Agrawal, *Embedded Sensor Systems*,
DOI 10.1007/978-981-10-3038-3_5

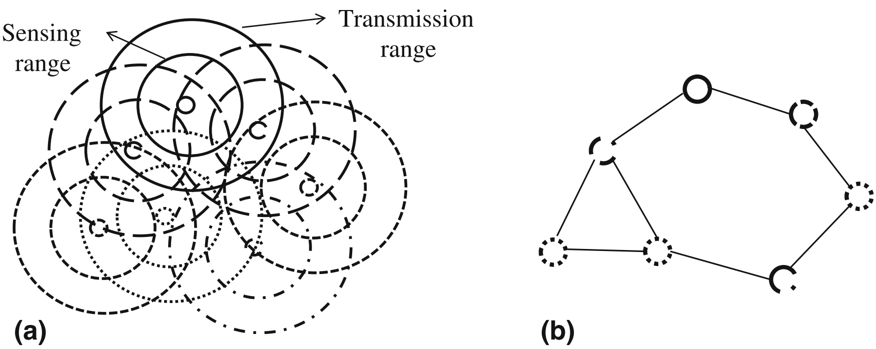


**Fig. 5.1** **a** Disk model of a transducer/SN and **b** graph model of a WSN

sensing current transmission from SN b, and SN c will not transmit. This is unnecessary as transmission from SN c to d does not interfere with current communication from SN a to SN b.

Each SN has a transmission range which determines its neighbors SNs with whom the SN can communicate. The $k$-neighborhood of a node $v$ is given by $\Delta_k(v)$, *and the* size of neighboring SNs is given by $v$:

$$\delta_k(v) = |\Delta_k(v)|. \tag{5.1}$$

Then, max $k$-neighborhood size in the network can be given by:

$$\delta_k = \max_v \delta_k(v). \tag{5.2}$$

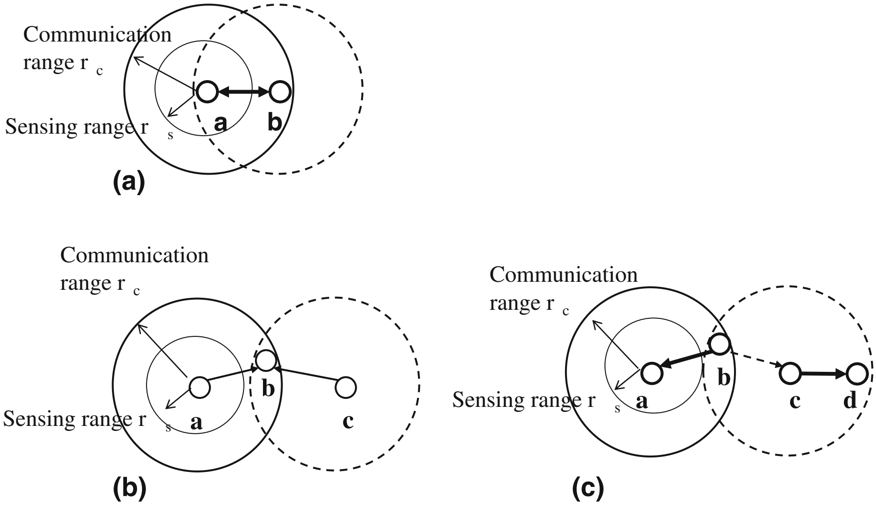


**Fig. 5.2** **a** Collision between 2 SNs, **b** hidden terminal problem and **c** exposed terminal problem

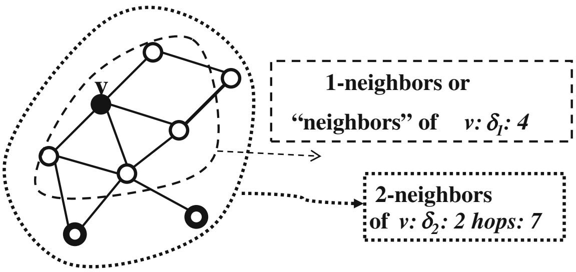


**Fig. 5.3** 1-hop and 2-hops neighbors of a reference SN *v*

Let *n* be the number of nodes in the network, then 1- and 2-neighbors of a SN *v* are shown in Fig. 5.3.

## 5.2 Clock and Signal Propagation in a WSN

Often, a hardware clock is present in a SN, with an oscillator generating pulses at a given frequency. A counter register is usually incremented after a fixed number of pulses, and only register content is available to software. A change rate of the register indicates resolution of the clock. $H_i(t)$ is defined as real time *t* of a SN *i*. Small letters (like *t*, *t*′) denote real physical times, while capital letters represent time stamps or anything else visible to SNs. A SN software local clock is usually determined as follows:

$$L_i(t) = q_i H_i(t) + f_i, \tag{5.3}$$

where $q_i$ is the drift rate and $f_i$ is the phase shift.

Time synchronization algorithms do modify $q_i$ and $f_i$, but not the counter register. If $C(t)$ is the value of a clock reported at time *t*, there exists $C'(t) = \mathrm{d}C(t)/\mathrm{d}t$ and $C'' = \mathrm{d}^2C(t)/\mathrm{d}t^2$ for $t \geq 0$. The difference between the time reported by two clocks is called offset. If time at two SNs clocks $C_1$ and $C_2$ are $C_1(t)$ and $C_2(t)$, respectively, the offset of the clock $C_1$ relative to clock $C_2$ at time $t \geq 0$ is $C_1(t) - C_2(t)$. The rate at which the clock progresses is called frequency. The frequency at time *t* of $C_1$ is $C_1'(t)$. The frequency ratio between two clocks is defined as clock ratio (a). The ratio of $C_1$ relative to $C_2$ at time *t* is $a = C_1'(t)/C_2'(t)$. The difference in the frequencies of two clocks is called as skew(d). The skew of $C_1$ relative to $C_2$ at time *t* is $d = C_1'(t) - C_2'(t) = aC_2'(t) - C_2'(t) = (a-1)C_2'(t)$. The drift of clock $C_1$ relative to the clock $C_2$ at time *t* is $C_1''(t) - C_2''(t)$ and basically represents the difference of change in the frequency rate. In a WSN, there are many sources of unknown, non-deterministic SNs, and the latency between time stamp of a message from a SN and its reception at another SN/BS ought to be considered. So, it is better if all SNs are synchronized together, and one SN can be taken as a reference for synchronization (Fig. 5.4).

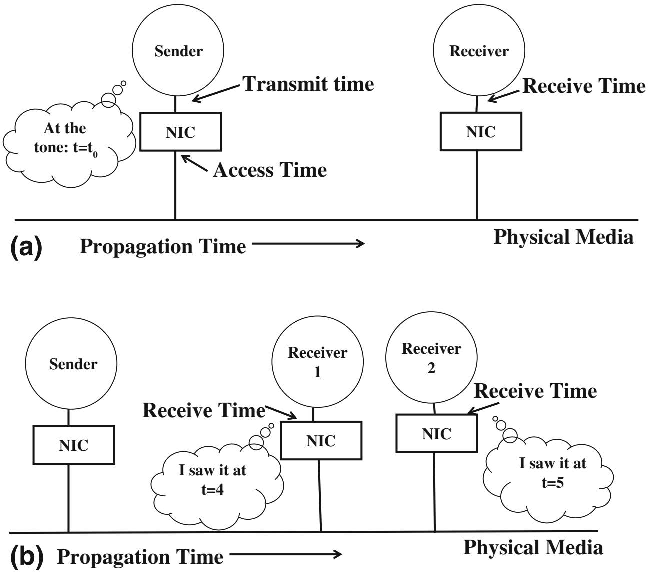


**Fig. 5.4** **a** Propagation of signals between two SNs and **b** sync reference broadcast

So, receiver-based synchronization (RBS) considers the time the sender SN reads its clock to when the receiver SN reads its clock and is defined as traditional critical path. RBS is only sensitive to the differences in receive time and propagation delays as shown in Fig. 5.5. RBS removes *send* and *access time* errors. Broadcast is used as a *relative* time reference, and each receiver synchronizes to a *reference packet. A r*eference packet is injected into the channel at the same time *for all receivers, and* time stamp is not there in the message. Any broadcast packet can be used as a reference such as RTS/CTS  packets and route discovery packets. It is natural to use BS clock as a reference as it usually broadcasts the query. The phase offset can be estimated as follows. For simplest case of a single pulse being sent to two receivers, the transmitter SN broadcasts reference packet and each receiver SN

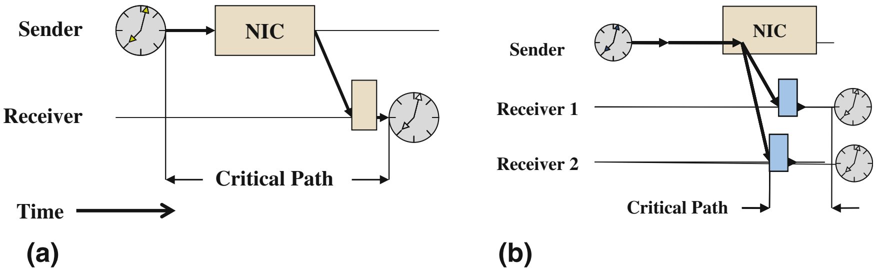


**Fig. 5.5** **a** Synchronization between a sender SN and a receiver SN [1] and **b** synchronization between a sender SN and multiple receiver SNs

records the time that beacon was received according to its *local clock*. Receiver SNs exchange their observations to have sufficient information to form a *local* (relative) timescale. However, global timescales are also important; simple case can be extended to many receivers assuming propagation delay to be zero, and no clock skew is present. In this way, receiver error follows *non*-determinism and is Gaussian in nature. Sending multiple messages always increases the precision. In such a case, a transmitter SN broadcasts *m* packets; each receiver SN records time the beacon was observed and exchange observed time; and receiver SN *i* computes phase offset to another receiver SN *j* as the *average* of the offsets implied by each pulse received by both SNs. This can be mathematically expressed as [1]:

$$\forall i \in n,\ j \in n: \quad \text{Offset}[i,j] = \frac{1}{m}\sum_{k=1}^{m}\left(T_{j,k} - T_{i,k}\right) \tag{5.4}$$

The packet reception time is shown in Fig. 5.6. The metrics for indicating effectiveness of synchronization mechanism are precision, longevity of synchronization, time and power budget available for synchronization , geographical span of the area to be monitored, and size and WSN topology. As receiver cannot predict when the packet will arrive and there could be clock skew, guard band is used in between. Time sync is critical at *many* layers in WSNs such as beam-forming, localization, tracking, distributed DSP, data aggregation , and caching. A clock in a computer at time t is given by an expression:

$$C(t) = k\int_0^t \omega(\tau)\,\mathrm{d}\tau + C(tl_0), \tag{5.5}$$

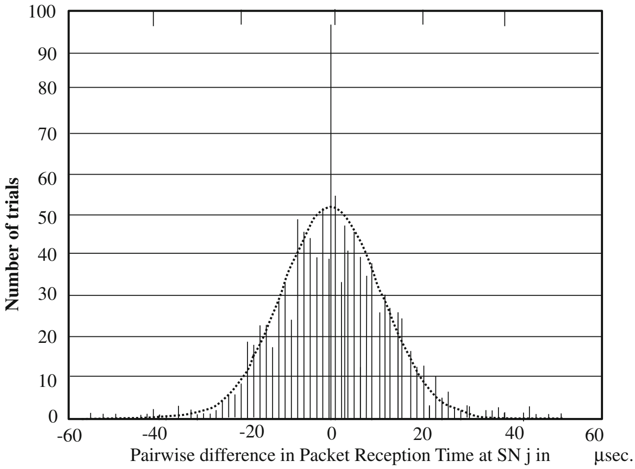


**Fig. 5.6** Pairwise difference in packet reception time at SN *j* in µsec [1]

where $\omega$ is frequency of the oscillator and $C(t_0)$ is the time of the computer click implemented based on a hardware oscillator. The computer clock (Fig. 5.7a) is an approximation of a real time $t$ as:

$$C(t) = q * t + b, \tag{5.6}$$

where $q$ is a clock drift and $b$ is an offset of the clock. In a perfect clock, rate $q = 1$ and offset $b = 0$.

The clock offset is defined as the difference between the local times of two SNs, and synchronization is required to adjust clock readings such that they match. The clock rate is the frequency at which a clock progresses and clock skew is the difference in frequencies of two clocks. The clock rate d$C$/d$t$ depends on temperature, humidity, supply voltage, age of quartz, etc., resulting in drift rate (d$C$/ d$t$ − 1). The sources of inaccuracies are primarily that SNs are switched on at random times and phase difference $\theta_i$ can be random. Actual oscillators do have random deviations from nominal frequency (drift, skew), and deviations are specified in ppm (pulses per million). The ppm value counts the additional pulses or lost pulses over the time of one million pulses at a nominal rate. The cheaper is the oscillator, and larger is the average deviation (Fig. 5.7b). For SNs, values between 1 ppm (one second every 11 days) and 100 ppm (one second every 2.8 h) are commonly assumed. Berkeley motes have an average drift of 40 ppm. The oscillator frequency depends on the time (oscillator aging) and environment (temperature, pressure, supply voltage, etc.). One-time synchronization is not sufficient for the time dependent as drift rates need to be resynchronized frequently. However, assuming stability over tens of minutes is considered reasonable.

General properties considered for time synchronization algorithms include physical time versus logical time, external versus internal synchronization , global versus local algorithms, synchronizing all SNs of a WSN synchronized or only a local neighborhood, absolute versus relative time synchronization, and hardware-versus software-based mechanisms. A GPS receiver is a hardware solution, but is too costly and energy-consuming in WSN SNs, and at least four satellites need to be on line of sight. It is important to consider whether a priori or a posteriori synchronization is needed after an interesting event, or whether a deterministic v/s

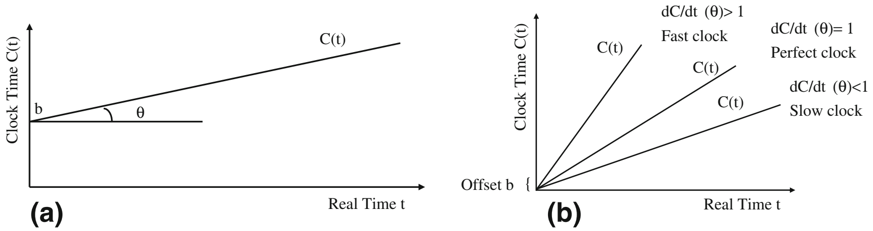


**Fig. 5.7** **a** Approximation of a computer clock and **b** computer clock parameters

stochastic precision bound is desirable. It should be noted that a backward jumps of local clocks be avoided and sudden jumps need to be avoided.

The performance metrics that ought to be considered are precision, energy costs, and fault tolerance. The precision indicates maximum synchronization error for deterministic algorithms, error mean/standard deviation/quintiles for stochastic values. The energy costs include number of exchanged packets, computational costs, and memory requirements. The maximum drift rate $\rho$ given by manufacturer (typical 1–100 ppm) guarantees that

$$1-\rho \leq \frac{\mathrm{d}C}{\mathrm{d}t} \leq 1+\rho. \tag{5.7}$$

The drift rate causes clocks to differ even after synchronization , and two synchronized identical clocks can drift from each other at rate of at most $2\rho_{\max}$. To limit relative offset to $\delta$ seconds, the resynchronization interval $\tau_{\text{sync}}$ must meet the requirement:

$$\tau_{\text{sync}} \leq \frac{\delta}{2\rho_{\max}}. \tag{5.8}$$

A fundamental building block of time synchronization is when to trigger resynchronization, what should be periodically, and should it be due to an external event? Estimating clocks of other SNs with exchanged packets and correcting and adjusting its own clock accordingly. Determining which SNs need to be synchronized with other SNs is an interesting challenge. What are the constraints for time synchronization  and whether an algorithm can scale to larger WSNs of unreliable SNs? The precision requirements could vary from ms to tens of seconds. The use of extra hardware such as GPS receivers is mostly not an option. The SNs have no or very low mobility ; often, there are no fixed upper bounds on packet delivery times due to MAC delays, buffering, etc.; propagation delay between neighboring nodes is negligible; and manual node configuration is not an option. The basic idea is that substantial energy costs are associated with frequent resynchronization if all SNs are kept synchronized all the time. This is specially overkill if a WSN becomes active very rarely. When a SN observes an external event at time t, it stores its local time stamp $L_i(t)$, achieves synchronization with neighbor SN/sink node, and converts $L_i(t)$ accordingly.

Synchronization can be implemented in many different ways, and challenges for time synchronization  are the environmental effects, energy constraints, wireless medium, mobility characteristics, etc. It could be either external or internal. External synchronization between a sender and a receiver and at least one node must have access to the external time scale. In internal receiver/receiver synchronization, no external timescale is used while nodes must agree on common time. For example, in Fig. 5.8a, when SN A senses an event, it starts counting time with its clock. SN A sends SN B a message regarding the event, and a time stamp includes how long the time has elapsed since the event. SN B estimates the

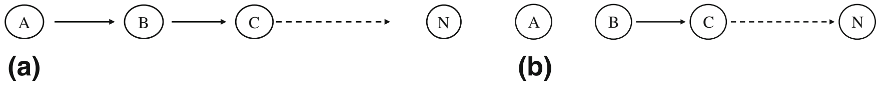


**Fig. 5.8** **a** Synchronization after an event and **b** message forwarding for synchronization after an event

transmission delay to the time stamp and continues counting the time. Messages are forwarded in the same fashion as A → B to SN N (Fig. 5.8b). N is able to recover the time by looking at the time stamp. The assumption is that the maximum clock skew is known, and the link can survive long enough so that a synchronization message can be sent after the application message (for given maximum drift rate $\rho$) as:

$$1-\rho \leq \frac{\Delta C}{\Delta t} \leq 1+\rho, \tag{5.9}$$

where $\Delta C$ is the drift in time $\Delta t$.

In sender/receiver synchronization, a receiver SN synchronizes its clock with a sender SN and pairwise synchronization involves synchronization of a single receiver SN to a single sender SN. Sources of inaccuracies [2] include MAC delay, interrupt latencies upon receiving packets, delays between packet interrupts and time-stamping operation, and delay in operating system and protocol stack. Improvement is possible if its packet is time-stamped after the MAC delay, immediately before transmitting the first bit. This removes the delay between packet interrupts and time-stamping from the uncertainty, and leaves only interrupt latencies. WSN-wide synchronization involves who synchronizes with whom to keep the whole WSN synchronized. The classical Network Time Protocol [3] belongs to this class. Lightweight Time Synchronization synchronizes the clocks of all SNs to one reference clock (e.g., equipped with GPS receivers) and is based on sender/receiver technique. A WSN-wide synchronization requires minimum spanning tree construction with reference node as root and synchronizes them in step by step mechanism. Here, a reference node R triggers construction of a spanning tree, it first synchronizes its neighbors, the first-level neighbors synchronize second-level neighbors, and so on. Different distributed algorithms for construction of spanning tree can be used, e.g., DDFS (distributed depth-first search). The cost consists of costs for construction of spanning tree and synchronizing two nodes costs 3 packets, while synchronizing $n$ nodes costs $\sim 3n$ packets. The other alternative is to use distributed multi-hop synchronization by not having an explicit spanning tree, but each SN knows identity of reference node(s) and routes to them. Minimum spanning tree is implicitly constructed as SNs need to know shortest route to the closest reference node. Other sender/receiver-based protocols are similar except the way spanning tree is constructed and how and when time stamps are incorporated.

## 5.3 Localization of a SN

Localization simply means determining where a given SN is physically located within a WSN. This process is needed to identify the location at which sensed reading originated from, and most applications use communication protocol that queries about geographic area, instead of identity of a SN. The purpose of a WSN is to monitor and report events take place in a particular area. Hence, the main parameters define how well the network observes a given area coverage. Localization techniques can be classified as shown in Fig. 5.9.

Coordinate-Free Localization is based on received signal strength indication (RSSI) measurement as strength of radio signal reduces with square of distance from the source that can indicate the distance of a SN from the source SN. But, as the signal travels through open space and radio propagation is highly non-uniform as wall, furniture, grass, asphalts, etc. absorb and reflect radio waves. The received signal usually contains noise in the order of several meters. The accuracy can be enhanced by either modifying measured distance or combining RSSI with some other parameters. RSSI is measured in dBm, which is 10 times the logarithm of the ratio of the received power ($P$) and the referenced transmitted power, and the equation leads to a straightforward indicator of distance estimation.

RSSI is known to be an inadequate estimator for determining the distance as the link quality between SNs behaves very inconsistently in estimating. Indoor is even worse (Fig. 5.10) as multi-path fading occurs due to reflection, refraction, and scattering of radio waves by structures inside a building even if multi-path medium is well characterized and modeled. Experimental results [4] have shown (Figs. 5.11 and 5.12) that RSSI estimates with imperfect links are difficult as RSSI patterns are observed to be complex. A slightest movement affects the received value significantly. RSSI serves at best as bounds for the distances. In fact, a single RSSI value can represent many distant locations in the room, where the same value is obtained.

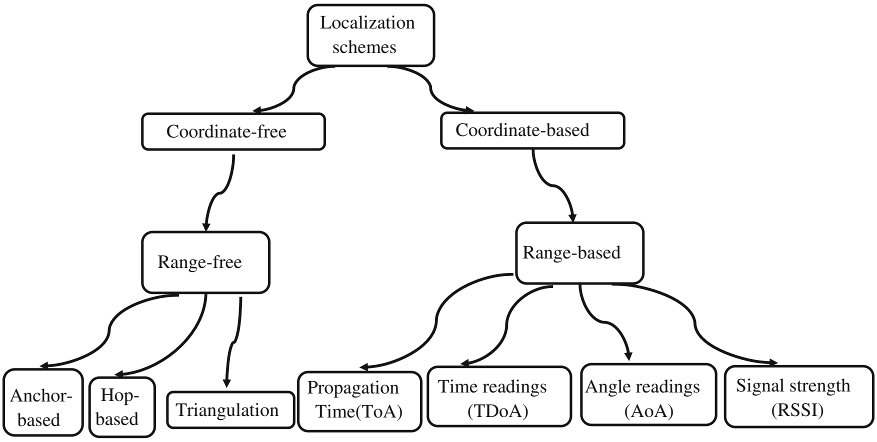


**Fig. 5.9** Characterizing localization techniques for SNs

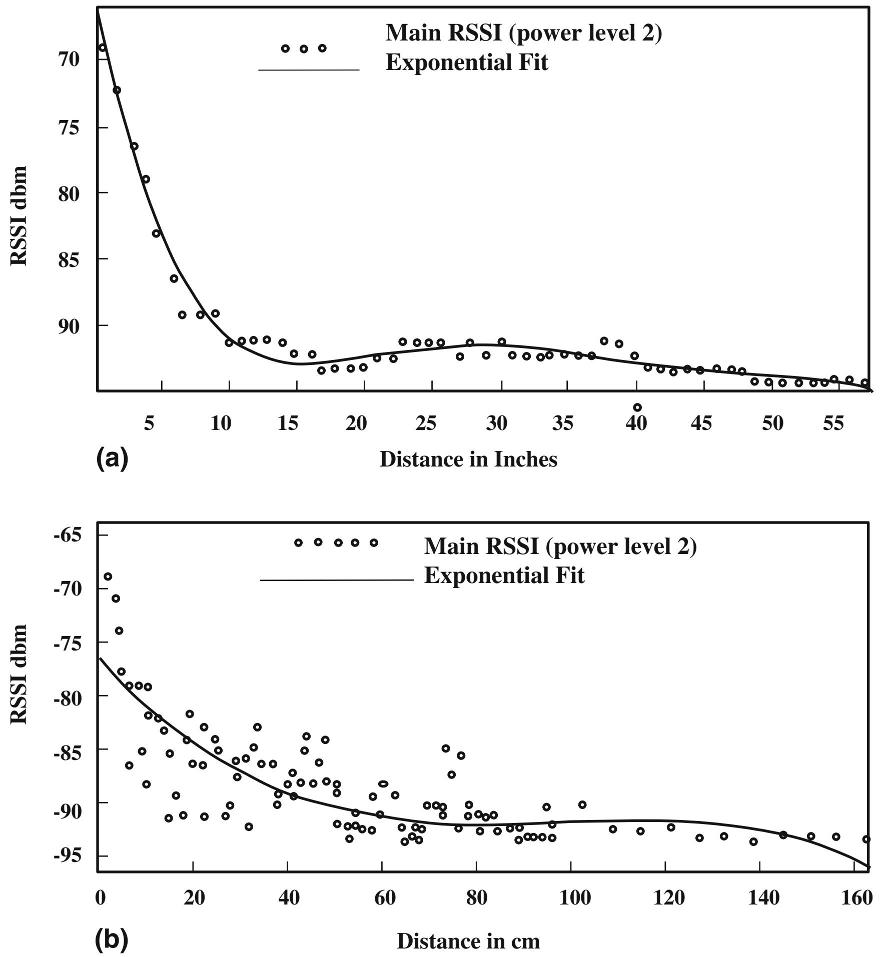


**Fig. 5.10** **a** Outdoor variation of RSSI and **b** indoor variation of RSSI [4]

Algorithms may fail, since strongly nonlinear distribution of RSSI makes this process error-prone. RSSI and the sender SN-announced power level do not necessarily match. The environmental factors (temperature and humidity) have been shown to interfere with RSSI readings as well. The link quality indicator (LQI), which is effectively a measure of chip error rate commonly available in commercial SNs, is observed to be highly correlated with both RSSI and with packet reception rate. Therefore, a combination of RSSI + LQI seems better suited for indoor localization. Distance estimation system has been proposed [4] that uses only the RSSI and LQI values. The test hypothesis was verified through experimental setup for both indoor and outdoor experiments. RSSI and LQI values were measured using Crossbow's TelosB mote (TPR2420CA) that has an integrated IEEE 802.15.4/ZigBee-compliant RF transceiver (CC2420 chipset) by varying distances

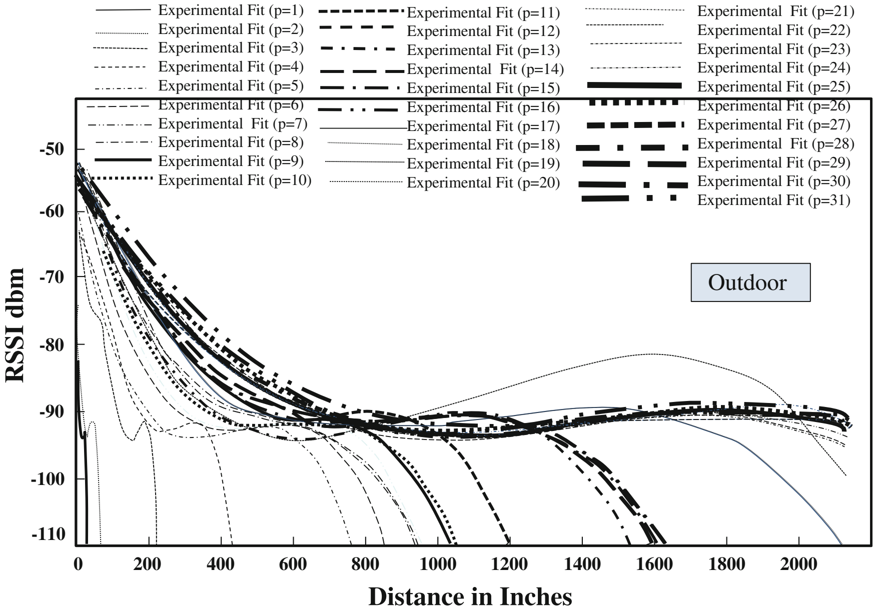


**Fig. 5.11** Outdoor variation of RSSI at different power levels [4]

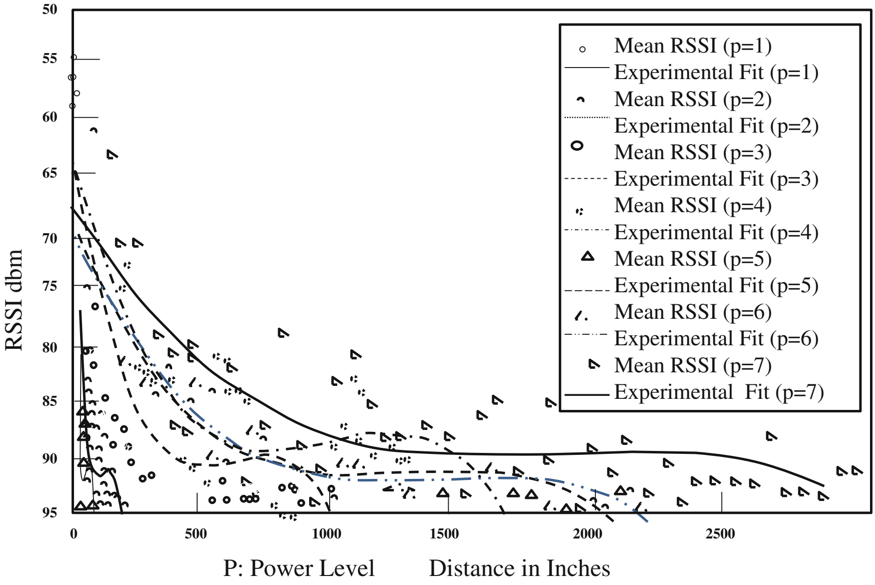


**Fig. 5.12** Indoor variation of RSSI at different power levels [4]

and power levels, and the results are summarized in Fig. 5.12. For each power level and distance, the mean of the RSSI at the receiver SN is measured that is observed to fit an exponential curve data. −93 dBm RSSI is observed at the distances [10, 13, 36, 37]. Figure 5.13a shows the number of elements when experiment performed indoor just for RSSI = −100 dBm measurement. When RSSI is combined with LQI, the membership reduces drastically as shown in Fig. 5.13b. Thus, a combination of (RSSI + LQI) seems better suited for indoor localization.

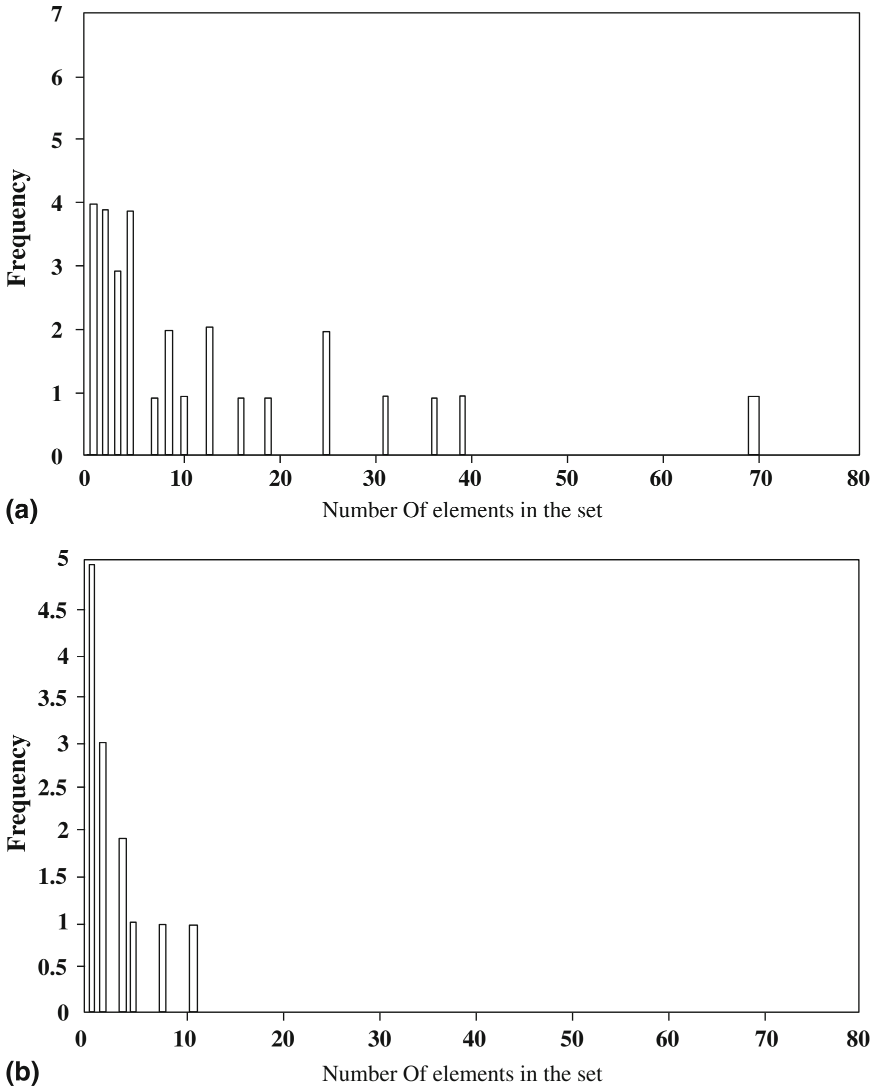


**Fig. 5.13** **a** Number of elements in RSSI set for indoor and **b** number of elements in RSSI + LQI set for indoor [4]

In a realistic WSN, RSSI is still used as a measure of distance. Assuming two SNs A and B can communicate with each other with distance less than $R$, then the hop count $h_{ij}$ can be used to compute internode distance between A and other SNs. The distance $d_{ij}$ between two SNs at $h_{ij}$ hops away will be less than $h_{ij} * d$ (Fig. 5.14), and the distance covered by one hop is:

$$d = R\left(1 + \mathrm{e}^{-n} - \int_{-1}^{1} \mathrm{e}^{-\frac{n}{\pi}}(\arccos\, t - t\sqrt{1 - t^2})\,\mathrm{d}t\right), \tag{5.10}$$

Triangulation is one scheme to determine the location by measuring angles from known points. Trilateration determines the location by measuring distance between reference points, while multilateration determines the location by measuring time difference of signal from reference points. Many range-free localization algorithms for the WSNs depend on trilateration algorithm and employs few SNs as reference anchors (Fig. 5.15a), assuming their position to be known. Other SNs use the information from the anchor nodes to estimate their own position (Fig. 5.15b) as:

$$\begin{bmatrix} (x - x_1)^2 + (y - y_1)^2 \\ (x - x_2)^2 + (y - y_2)^2 \\ (x - x_3)^2 + (y - y_3)^2 \end{bmatrix} = \begin{bmatrix} d_1^2 \\ d_2^2 \\ d_3^2 \end{bmatrix}, \tag{5.11}$$

leading to:

$$\begin{aligned} &2x(x_i - x_1) + 2y(y_i - y_1) = d_1^2 - d_i^2 + x_i^2 - x_1^2 + y_i^2 - y_1^2 \\ &(i = 2, 3) \end{aligned} \tag{5.12}$$

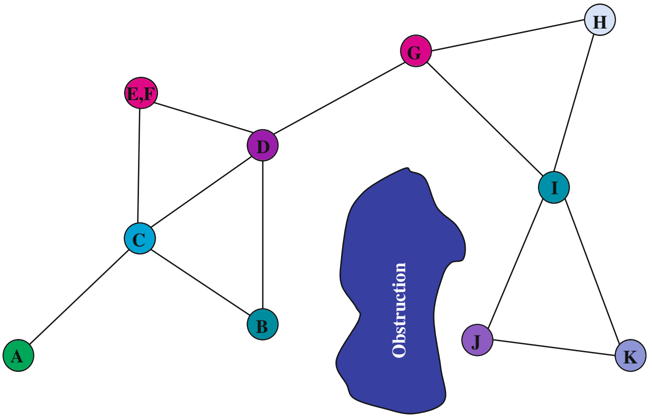


**Fig. 5.14** Number of elements in RSSI set for indoor [5]

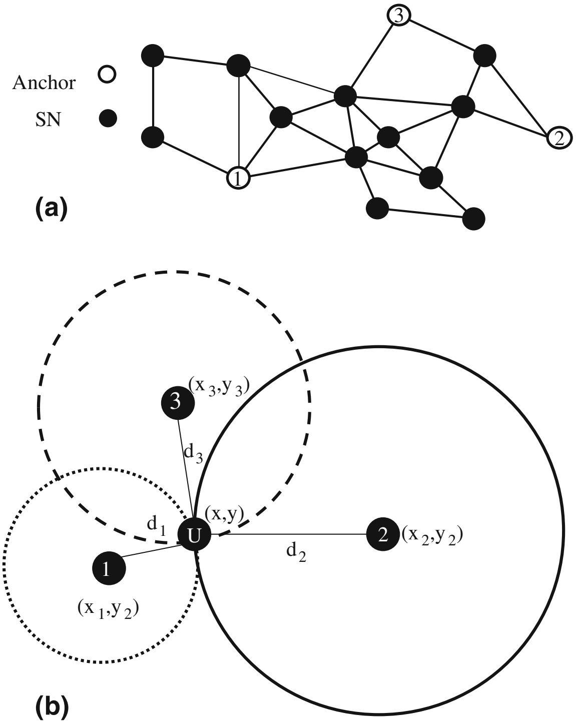
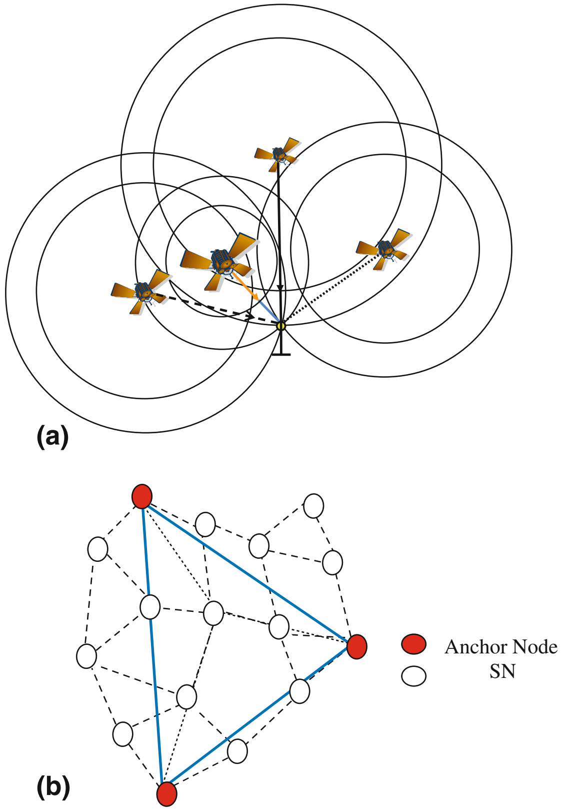


**Fig. 5.15** **a** Trilateration in a WSN with selected anchors and **b** trilateration algorithm with 3 anchors

Besides RSSI that has multi-path fading, background interference, and irregular signal propagation , there are many ways to measure the distance between a SN and anchor SNs. These include time of arrival (ToA) (time of arrival ), GPS with expensive in hardware and energy-consuming, TDoA (time difference of arrival ), and AoA (angle of arrival). AoA is based on estimating relative angles between neighbors and requires additional hardware. It could be expensive to deploy in large WSNs. One way of measuring ToA with GPS is to use satellite constellation of at least 4 satellites with atomic clocks (Fig. 5.16a). As satellites broadcast precise time, the ToA is an estimated distance to satellite and trilateration determines location of a SN. This process is expensive, and it is rather impractical to use the absolute address. Locations can be determined by using AoA by using antenna array to measure the direction of neighbors. Special landmarks or anchors usually have compass and GPS capability. They broadcast location and flood beacons, and update bearing along the way. Once beacons from three landmarks are known, the position of the SN can be calculated as shown in Fig. 5.16b.

In a basic APIT (approximate point-in-triangulation test) Scheme [7], anchors are location aware SNs in the WSN. APIT employs area-based approach to isolate triangular regions between beaconing nodes. Once the area is known, the CoG (center of gravity) calculation is performed for the location of the SN as illustrated in Fig. 5.17a. In a perfect PIT (point-in-triangulation test) [8] scheme (Fig. 5.17b),

**Fig. 5.16** **a** Angle of arrival in a WSN with selected satellites as anchors [6] and **b** time of arrival in a WSN with selected anchors



given anchor nodes A, B, and C, if a SN M is inside the triangle ABC, when M is shifted in any direction, the new position must be nearer to (further from) at least one anchor node A, B, or C. If the SN M is outside the triangle ABC; when M is shifted, there must exist a direction in which the position of M is further from or closer to all three anchors A, B, and C. If there exists a direction such that a point adjacent to M is further/closer to points A, B, and C simultaneously, then M is outside of ABC. Otherwise, M is inside ABC. A perfect PIT test is infeasible in practice. Experiments show that the received signal strength is decreasing in an environment without obstacles. Therefore, further away a node is from the anchor, and weaker will be the received signal strength.

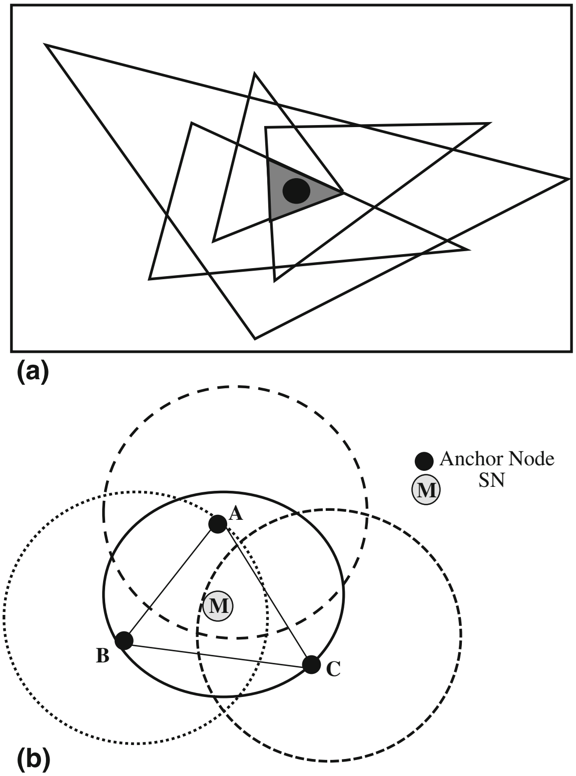


**Fig. 5.17** **a** Approximate triangulation scheme and **b** point-in-triangulation scheme

## 5.4 Conclusions

As a large number of SNs ought to be deployed in a typical WSN application, synchronization among SNs has become an important aspect. This is especially critical if SNs are forced to go to sleep mode and are awakened when SN or neighboring SN wants to transmit data to BS in a multi-hop fashion. Both transmitting SN and receiving SN need to be awake and synchronized for any successful information transfer. In a WSN, 1-hop synchronization is easier to implement while multi-hop synchronization is relatively complex to achieve, even though that has to be eventually achieved directly or indirectly. As received data determine the presence of an event, the localization is needed to determine where the desired information is coming from, indicating the presence of an event.

It is important to determine how many events (moving objects) have been detected, and what is the direction and speed of the moving object? Thus, synchronization and localization are two issues are complimentary to each other and are essential for successful operation of a large WSN.

## 5.5 Questions

Q.5.1. How can time synchronization be propagated across the network by using a spanning tree favoring direct connections with reliable delays?

Q.5.2. Assuming that the clock of each CH is synchronized with the BS, what are the limits of maximum and minimum errors? Can you determine similar value for a sensor node?

Q.5.3. What is the key idea of the interval method in a WSN?

Q.5.4. Why does synchronization remain especially challenging problems in a WSN?

Q.5.5. Assume that two nodes have a maximum drift rate from the real time of 200 ppm each. Your goal is to synchronize their clocks such that their relative offset does not exceed 2 s. What is the necessary resynchronization interval?

Q.5.6. A network of five nodes is synchronized to an external reference time with maximum errors of 1, 3, 4, 1, and 2 time units, respectively. What is the maximum precision that can be obtained in this network?

Q.5.7. Node A sends a synchronization request to node B at 3150 (on node A's clock). At 3250, node A receives the reply from node B with a time stamp of 3120. (a) What is node A's clock offset with respect to the time at node B (you can ignore any processing delays at either node)? (b) Is node A's clock going too slow or too fast? (c) How should node A adjust the clock?

Q.5.8. Three beacons are located at $a = (1, 1)$, $b = (1, -1)$, and $c = (-1, 1)$. The received powers from nodes $a$, $b$, and $c$ are 1.2, 1.5, and 1.7, respectively. Calculate the unknown position of the receiver through a weighted centroid computation.

Q.5.9. Assume that five reference nodes are known at (0, 3, 0), (6, 0, 0), (3, 4, 0), (−4, −3, 0), and (0, 0, −8), respectively; also, $t12 = 0$ s, $t13 = 1$ s, $t14 = 0.7$ s, $t15 = 0.7$ s, and $t16 = 1.7$ s. The velocity of propagation is $v$. (a) Find the unknown location ($xt$, $yt$, $zt$). (b) Now assume that the propagation speed is known to be 8.7 m/s. Find the unknown location ($xt$, $yt$, $zt$).

Q.5.10. How is localizing and tracking of moving objects an essential capability for a WSN?

Q.5.11. What is the goal of localization or tracking in obtaining a good estimate of the target state?

Q.5.12. Can you have joint time synchronization and localization of an unknown node in a WSN?

## References

1. J. Elson, slides, cs.uccs.edu/~cs526/mote/mobilcome/Mobicom-Tutorial-3-MS.ppt.
2. Noh Kyoung-lae Noh, E. Serpedin, and K. Qaraqe, “A New Approach for Time Synchronization in Wireless Sensor Networks: Pairwise Broadcast Synchronization,” IEEE Wireless Communications, vol. 7, no. 9, pp. 3318–3322, Sept. 2008.
3. Network Time Protocol (Version 3) Specification, Implementation and Analysis, https://tools.ietf.org/html/rfc1305.
4. Madhanmohan Raju, Talmai B Oliveira, and Dharma P Agrawal, *“A practical distance estimator through distributed RSSI/LQI processing - An experimental study,” IEEE International Conference on Communications,* WS-SCPA (2nd IEEE International Workshop on Smart Communication Protocols and Algorithms), *Ottawa, Canada, June 15, 2012, pp. 8155–8159.*
5. Asma Mesmoudi, Mohammed Feham, and Nabila Labraoui, “Wireless Sensor Networks Localization Algorithms: A Comprehensive Survey,” International Journal of Computer Networks & Communications (IJCNC), vol.5, no. 6, pp. 45–64, November 2013.
6. J. Bachrach and C. Taylor, “Localization in Sensor Networks,” Handbook of Sensor Networks: Algorithms and Architectures, Wiley-Interscience, 2005.
7. Waltenegus Dargie and Christian Poellabauer, Fundamentals of Wireless Sensor Networks: Theory and Practice, John Wiley & Sons Ltd., 2010.
8. M. Venkatakrishnan and A. K. Yuvashree, “Survey on Range Free-Localization Schemes Based on Anchors Count in WSN Model,” IJIRST–International Journal for Innovative Research in Science & Technology, vol. 1, no. 9, February 2015.