Research and development on aspects of clock synchronization in a wireless sensor network

Zeyad Ghaleb Al-Mekhlafi 1, Zurina Mohd Hanapi 2, Mohamed Othman 3, and Zuriati Ahmad Zukarnain 4

*1 PhD student, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, ziadgh2003@hotmail.com
2, 3, 4 Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, zurinamh@upm.edu.my; mothman@fsktm.upm.edu.my; zuriati@fsktm.upm.edu.my

Abstract

Sensor networks are an example of a wireless network that needs each sensor to execute events in synchronization. This synchronization is necessary to coordinate power cycles to conserve energy and to ensure the proper functioning of sensors that measure time-sensitive events. Clock synchronization is important to the operation of each distributed wireless sensor network (WSN) because it sets a common time limit to different sensor nodes. This paper aims to review the fundamental aspects of the deafness problem, which is the main problem of clock synchronization for WSNs. Deafness occurs when sensor nodes could not receive and transmit simultaneously. Synchronization can be achieved based on data collected from WSNs. Synchronization-based data gathering is geared towards maintaining and optimizing energy efficiency for time data collection in sensor nodes. This study serves as a useful source on clock synchronization to assist WSN researchers and novices to gain a better understanding of the deafness problem and to promote effective designs and systems that address this problem.

Keywords: Wireless Sensor Networks (WSNs), Clock Synchronization, Deafness Problem

1. Introduction

Wireless sensor networks (WSNs) have limited processing power and transmission range [1]. Routing protocol design in WSN is a challenging issue given restricted WSN capacities (i.e., insufficient power or energy, total failure, limited processing power, restricted sensing region, and lower bandwidth), which may result in sensor node breakdown [2]. Node failure results in a sensor network inability to perform regular processing and transmit information to the target [3]. Each routing protocol algorithm sends data from the sources to the targets and is expected to increase network exposure even as the propagation value decreases [4]. In general, WSN is a collection of self-directed devices that are associated wirelessly. Sensor networks are an example of wireless networks that need every sensor to execute events in synchronization. This synchronization coordinates power cycles to conserve energy and ensures the smooth operation of WSNs that calculate time-sensitive events.

Clock synchronization is an important issue in the operation of any distributed WSN. Synchronization sets the same time limit for different sensor nodes, which unifies functions for video and voice data, organizes different wakeup or sleep node scheduling schemes, and ensures time-based channel distribution [5]. Clock synchronization has several advantages over unsynchronized systems [6] and is often inherently assumed to facilitate certain techniques and algorithms on physical and medium access control (MAC) layers. From the physical layer perspective, slot synchronization enables advanced cooperative transmission technologies. From a MAC layer perspective, slot synchronization is a form of coordination among nodes to enhance power efficiency and throughput.

Deafness problem is the main problem of clock synchronization in a WSN, which is associated with the ready-to-send (RTS)/clear-to-send (CTS) packet based on protocols. Deafness occurs when a transmitter sends a control packet to initiate propagation, and the destination is tuned to another channel. If the transmitter does not receive any response after sending multiple requests, it may conclude that the
receiver is no longer reachable [7]. Overlapping communications and deafness in existing WSNs occur when a broadcast communication medium is used because normal wireless transmitters cannot receive messages when they are in transmission mode. Deafness can be avoided by distributing the synchronized message through a random offset, even as the detailed offset is transmitted through the message. Afterwards, the receiver can immediately reestablish synchronization and alter the time as a result of the received offset costs.

2. Wireless Sensor Network

Various researchers have provided different definitions of WSN. Ramirez et al. [8] defined WSN as a collection of autonomous devices or nodes that are connected wirelessly. Sharma et al. [9] described WSN as a group of thousands of tiny sensor nodes that can perform wireless communication, limited calculation, and sensing.

Hanapi [10] stated that WSN can be a heterogeneous sensor network that consists of many low-cost and low-power sensor nodes that are more likely deployed at fixed locations. These sensor nodes can communicate with one another through radio frequency (RF), sense and relay sensor data to other users, and compute physical attributes (e.g., pressure, temperature, motion, sound, and vibration). Stojčev et al. [11] explained that WSNs are large-scale sensor networks that monitor and observe various aspects of the natural world. An excerpt from the Defense Advanced Research Project Agency in Bala [12] states that sensor node networks use many devices that can compute, sense, and communicate through additional devices to compile local data and formulate conclusions about the physical world. Bala [12] also mentioned that according to the United States National Research Council, sensor node networks are composed of a large number of sensors that are commonly used in mechanical and electrical systems to manage (i.e., effect) and observe (i.e., sense) almost all aspects of the natural world.

A WSN comprises a sensor, node, base station, gateway, and coordinator. Figure 1 illustrates the WSN elements.

Table 1 provides the definition of each element.

<table>
<thead>
<tr>
<th>WSN Elements</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensors</td>
<td>Sensors are the heart of network devices. They obtain data from the medium and convert the data into wireless signals.</td>
</tr>
<tr>
<td>Nodes</td>
<td>Nodes are the basic units of WSN. These nodes obtain data from the sensors and relay the information to the base station [3, 13, 14]. Nodes are small devices that have a few kilobytes of memory, MHz processors, a radio scope of a few meters, and one or two batteries.</td>
</tr>
<tr>
<td>Gateway</td>
<td>The gateway acts as an entrance or a proxy server and firewall to another network. Gateways facilitate intelligent wireless sensor network network-data (TCP/IP) network</td>
</tr>
</tbody>
</table>
A sensor often has limited processing and sensing capabilities. An essential task over a wide physical region could be accomplished through instant collaboration between the information aggregated across the entire network and sensors[14].

Certain aspects of WSNs should be addressed, such as self-management for auto-configuration, efficient power management to decrease energy use and adjustment for varying topologies, and clock synchronization to ensure a common clock/time frame throughout the whole WSN. In this paper, we consider the clock synchronization issue and review the fundamental aspects of the deafness problem in WSNs.

3. Clock Synchronization in WSNs

Each sensor node in WSNs has its own clock. Clock synchronization provides a common clock/time frame for widely distributed sensors. However, this task is not easy to accomplish because of the unique properties of WSNs. A universal time is normally unavailable in WSNs. Therefore, traditional clock/time center-based synchronization methods or tools cannot be applied directly on each of the services at the sensor nodes (e.g., sensing, routing, group management, localization, time synchronization, power management, and medium access control). Moreover, sensors often have limited processing and sensing capabilities. Therefore, sensors have to cooperate to perform a task over a wide physical region, which can be achieved with the use of information aggregated across the entire network and sensors. Hence, a low-overhead method and accurate clock synchronization are ideal for sensor-based applications. In national schemes, clock synchronization is unnecessary because clock confusion does not exist. By contrast, in disseminated systems such as WSNs, no universal memory or time exists[11].

Clock synchronization in WSNs has attracted extensive attention because it is a crucial issue in the operation of WSNs [16]. It unifies different functions, such as video and voice data from dissimilar sensor nodes, wake/sleep scheduling for nodes, and time-based channel sharing [17, 18]. A consistent clock/time is important for sense functions to ensure precise timestamping of sectioned information [4]. Clock synchronization is a complex problem that can be resolved by using the computer system of a distributed scheme.

3.1. Importance of Clock Synchronization

Each sensor node maintains a local time generated by its own clock (its own concept of time). Different factors make flexible and robust clock synchronization especially important. Time in sensor nodes is typically conserved by a particular sub-scheme, as shown in Figure 2 [19].

![Figure 2. Basic Block Diagram of Clock Elements and Associated Timer Hardware.](image)

The time driver stimulates the resonating component that eventually resonates and filters at a certain frequency. Software can use this hardware counter for time measurement and timers. The hardware
counter utilizes a signal to increase a counting register at regular intervals.

For any two clock $C_a$ and $C_b$, our study will propose the following terminologies, as shown in Table 2, which are consistent with definitions given in [18, 20-22]:

<table>
<thead>
<tr>
<th>Clock Terminology</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>The time of a clock is assumed by the function $C_p(t) = t$</td>
</tr>
<tr>
<td>Frequency</td>
<td>The frequency at time $t$ of the clock $C_a$ is $C_a'(t)$.</td>
</tr>
<tr>
<td>Offset</td>
<td>Time offset is the dissimilarity among the time reported by the actual time and a clock. The offset of the clock $C_a$ is assumed by $C_a(t)$. The offset of clock $C_a$ relative to clock $C_b$ at time $t \geq 0$ is assumed by $C_a(t) - C_b(t)$.</td>
</tr>
<tr>
<td>Skew</td>
<td>The skew of a clock $C_a$ relative to clock $C_b$ at time $t$ is $C_a'(t) - C_b'(t)$. Whether the skew is surrounded by $p$, then as per eq. (1), clock costs can diverge at a rate that ranges from $1-p$ to $1+p$.</td>
</tr>
<tr>
<td>Drift</td>
<td>The drift of clock $C_a$ is the second derivative of the clock cost with respect to time, namely, $C_a''$. The drift of clock $C_a$ relative to clock $C_b$ at time $t$ is $C_a''(t) - C_b''(t)$.</td>
</tr>
</tbody>
</table>

For existing algorithms and applications that use WSNs, we aim to know more about time synchronization, including classification according to the relative arrangement of actions that occurred in different sensor nodes, the time of the day when an event occurred in a particular sensor node, and the time period among two actions that occurred in various sensor nodes. For the system design of applications and algorithms, clock synchronization sets a common clock/time throughout a distributed system and ensures that WSNs can perform basic operations. These basic operations, which do not require the same sensors to work within a common clock/time frame, are data fusion, power management, and transmission scheduling. Table 3 provides the definitions of these operations.

<table>
<thead>
<tr>
<th>Fundamental operations</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Fusion</td>
<td>Data fusion is a fundamental operation in all disseminated WSNs for integrating and processing gathered data. WSNs typically span vast geographic regions and consist of many nodes because of the limitations of individual sensor nodes. One single sensor cannot capture all information, and thus, information from all sensors should be obtained. Data fusion requires several or all nodes in the WSN to have a common time scale. An event can be monitored simultaneously by multiple sensors. Information at dissimilar sensors can be combined to contain extra data. Such information integration requires several or all sensors to have a common time scale. This condition is necessary when the environment under surveillance is time varying. In entity tracking, every sensor node discovers a moving entity when it enters the sensor’s vicinity. The entity can be tracked by cross-referencing its time and position as recorded by sensors along its path. The recorded time is accurate when the times of the sensors are synchronized [23-26].</td>
</tr>
<tr>
<td>Power Management</td>
<td>Power management is an essential factor for WSNs because sensors node are typically left unattended and are battery powered. In addition, they do not undergo regular servicing or battery changes. Most fundamental operations in WSN will adopt wake/sleep protocols to conserve energy wherever certain sensors enter a low-power sleep mode or switch off when their neighboring sensors are on duty. Therefore, network-wide clock synchronization is important because it can ensure time synchronization precision and efficient power cycling [27].</td>
</tr>
<tr>
<td>Transmission Scheduling</td>
<td>The MAC layer has many protocols that require synchronization. For example, time division multiple access (TDMA) accesses and permits multiple devices to distribute access to a common communication medium [28]. One transmission period is classified into multiple slots in TDMA to allow transmission without collisions or interference, and every slot is allocated to only one sensor node in a suitable area. Each sensor node is enabled to transmit only throughout the dedicated time slot [27]. Such protocols are valid only in a synchronized network. So in the Transmission Scheduling there many problems happened such as deafness problem, idle listening problem, collision-free, hidden terminal problem and overhearing problem.</td>
</tr>
</tbody>
</table>
WSNs are fundamentally disseminated systems. Clock synchronization is a prerequisite in many procedures for disseminated systems (e.g., user transactions, system analysis and debugging, encryption, and seamless status updating protocols) [27]. Many tracking, security, and localization protocols order sensor node networks to timestamp their sensing steps and messages. Consequently, time synchronization among different nodes is essential for WSNs, and clock synchronization is a crucial issue in the design of energy-efficient WSNs [27].

3.2. Challenges to Clock Synchronization

In recent years, numerous algorithms/protocols have been designed to maintain synchronized clocks over computer networks. A flowchart of planned clock synchronization is presented in Figure 3. Unfortunately, in an actual wireless network, various component delays affect message delivery. Table 4 explains the causes of each contribution and indicates the variability of the delays and randomness [29, 30]. Ensuring time synchronization is more difficult than it appears to be.

![Figure 3. Planned Clock Synchronization.](image)

A communication propagation sequence should be created to approximate the relative time offsets and skews between nodes. Time synchronization in WSNs should be considered when eliminating the impacts of random delays from the technique communication propagations forwarded in WSN channels.

| Table 4. Definition of the component of delay effects, causes, and randomness. |
|--------------------------|-----------------------------------------------------------------|--------------------------|
| Definition               | Cause                                                                 | Randomness               |
| Send Time                | After transmitting the request, the operating system shows the time spent in constructing the communication at the application layer among delays. The sender also spends this time to synchronize communication and leave by this message to the network limit. This time is nondeterministic and can be over hundreds of milliseconds (ms) depending on the amount of system work. | Build synchronization message | Low               |
| Access Time              | Time spent to access channels when arriving at the MAC layer. This time is an important factor and is extremely variable depending on the precise MAC protocol, which differs from milliseconds (ms) to seconds depending on existing network traffic. | Locate message on medium according to the MAC protocol. | High              |
| Transmission Time        | The time spent to transmit a message at the physical layer (PHY). This time is approximately tens of milliseconds (ms) and is normally deterministic, which could be expected from the message length and radio speed. | Physical transmission of signal | Transmission Time |
| Propagation Time         | This is the time it takes for the message to be transmitted from the sender to the receiver across a wireless channel. This time is deterministic and, in general, is less than one microsecond (µs), which is approximately negligible compared with other delay mechanisms. | Physical propagation of signal | Propagation Time  |
| Receive Time             | The time at the application layer for the receiver or spent by the recipient to process the message and inform the host of its arrival. This is also the time spent to build and send the received message. This time can vary because of variable delays in usage system. | Process synchronization message | Receive Time      |
Delay components can be classified into random and fixed delays. Random delays rely on diverse network parameters (e.g., traffic and network status). Thus, it applies to different cases and has been modeled as random delays in WSNs that contain Gamma sharing, Gaussian sharing, exponential sharing, and Weibull sharing based on different applications and validations [31-33].

Fixed delays are typically unfamiliar and if they are not modeled properly, they will be considered a part of the clock offset, which results in less precise timing parameter estimation [34]. WSNs also have to deal with limited and non-rechargeable power resources in clock synchronization. Time synchronization contributes to energy consumption because of the great amount of energy used by radio propagations to transmit time data. RF requires 3 J to transmit 1 kb over a hundred meters, which is equal to the energy required to transmit to transmit to three million directions [35]. Thus, efficient synchronization algorithms can reduce communication overhead and computational power.

3.3. Fundamental Approaches to Clock Synchronization for WSN

Transmitting sensors are the basic components involved in time synchronization, which could be accomplished by transferring timing communications to the sensor nodes; these timing communications are timestamped. Fundamental clock synchronization approaches can be classified into three timing communication signaling approaches [5, 30], as shown in Table 5.

<table>
<thead>
<tr>
<th>Fundamental Approaches</th>
<th>Definition</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receiver–receiver synchronization</td>
<td>Sensor nodes perform their local clocks autonomously, but they contain data on the relative drift and offset of their clock to other clocks in the network [36].</td>
<td>RBS[37], HRTS[38], PBS[39].</td>
</tr>
<tr>
<td>One-way message dissemination</td>
<td>The easiest form of clock synchronization, this involves requesting messages or events. Determining whether event E1 occurred before or after a new event E2 is possible by using this model [40].</td>
<td>FTSP[41], SPS[42], NTP[43].</td>
</tr>
<tr>
<td>Two-way message exchange</td>
<td>This is a multifaceted form of synchronization in which the time of every sensor node is synchronized to the network [44].</td>
<td>TPSN[45], LTS[46].</td>
</tr>
</tbody>
</table>

3.4 Requirements of Clock Synchronization Schemes for WSN

Clock synchronization requirements can be considered metrics for evaluating clock synchronization designs for WSNs [47]. Compromises among the requirements of an efficient synchronization approach [29, 30] exist, as shown in Table 6. As a result, a single scheme may not satisfy all the requirements.

<table>
<thead>
<tr>
<th>Requirements of Clock Synchronization</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy Efficiency</td>
<td>The most significant issue is the limited energy resources for all protocols of sensor node networks. Sensor node networks use small, low-cost high performance batteries particularly in large-scale networks and when sensor nodes are in difficult-to-serve locations. Thus, synchronization designs should consider the limited energy resources of sensor nodes[30].</td>
</tr>
<tr>
<td>Scalability</td>
<td>WSNs contain many sensors. Thus, a perfect clock synchronization protocol must be able to adapt to and exhibit consistent performance despite a large number of nodes and/or a high-density network. Scalability cannot be achieved by structures that require global information such as routing table entries or addresses [48]. Therefore, such information should be limited, and infrastructureless protocols are ideal [49].</td>
</tr>
<tr>
<td>Robustness</td>
<td>Sensors node can shift in/out of every other communication range because of mobility, and</td>
</tr>
</tbody>
</table>
a sensor node can break down because of limited power. Furthermore, a sensor node network is normally left unobserved for long periods of time in probably hostile environments. To prevent sensor nodes from failing, the synchronization scheme must be valid and useful all throughout the network [50].

### Precision or Accuracy
The requirement for accuracy or precision may vary depending on the particular application and the function of synchronization. For some applications, a simple ordering of messages and events might suffice, whereas other require synchronization precision or accuracy in the order of some microseconds [51, 52].

### Lifetime
The clock synchronized between sensor nodes by a synchronization algorithm may be last or first depending on the operation time of the network [53].

### Scope
The synchronization design can provide a global clock base for local synchronization only between spatially close nodes or every node in the network. Scalability issues make global time synchronization difficult to accomplish or expensive (bandwidth usage and considering energy) in a large sensor node network. Moreover, a common clock/time base for many sensors is required to aggregate information from distant nodes, which requires global clock synchronization [18].

### Size and Cost
As previously noted, WSN nodes are small, low-cost devices. Thus, a synchronization method for sensor networks that considers the limited size and cost must be developed [30].

### Immediacy
In emergency detection, the sensor network needs to convert an event immediately to the aggregation node. For this type of application, the network could not tolerate any delay after such an emergency is identified. Immediacy can prevent the protocol designer from depending on too much processing when an event of interest happens, which requires that nodes be pre-synchronized at all times [54].

### 3.5. Clock Synchronization Protocols for WSN

Depending on network size, clock synchronization protocols can be classified into network-wide and pairwise synchronization.

- **Network-wide clock synchronization for WSN** concentrates on sensor nodes that are organized into a multi-hop network. This type of synchronization can generally be obtained by extending pairwise clock synchronization derived from diverse communication structures, for which perfect structures must be robust, low cost, and scalable. Network-wide clock synchronization also aims to develop a common time frame for a group of sensor nodes such that any two sensors have very similar clock readings [55].

- **Pairwise clock synchronization for WSN** concentrates on two neighboring sensor nodes that are in every other node’s communication range, for which perfect algorithms achieve precise clock synchronization and reduce random effects caused by communication delays through communication load and minimum computation. Pairwise clock synchronization for WSN in the presence of unknown exponential delays was also considered under a two-way message exchange mechanism [56].

### 4. Deafness Problem

The deafness problem is associated with the RTS or CTS packet in the IEEE 802.11 MAC protocol [57] and occurs when a transmitter sends a control packet to initiate a transmission while the destination is tuned to another channel. After sending multiple requests, if the transmitter does not receive any response, it may conclude that the receiver is no longer reachable [7].

Figure 4. Sensor Node that Explains the Deafness Problem. Source: [58]
Figure 4 shows a well-known deafness problem. Node A, which is a neighbor of node N, is unaware that N is in direct communication with another neighbor and attempts to communicate with N by sending a packet. Node N, which is a beam formed in a different direction, fails to hear the packet. Assuming congestion as the cause of failure, a backoff occurs before attempting retransmission. If data packets are large, then N remains engaged in communication to S for a long time, during which A may attempt multiple retransmissions, with each retransmission preceded by increasing backoff duration. When N finally completes its packet delivery and is ready to transmit a new one, A is highly likely to be counting down a larger backoff counter. Deafness occurs when node A continues to send RTS messages to node N, and node N is deaf to these messages. Node N does not reply to any CTS message sent by node A, which then goes into backoff mode [58]. So the deafness leads to longer delay, wasting energy, excessive packet drop, and channel access unfairness.

Message collisions and the deafness problem in existing WSNs occur because normal wireless transmitters cannot receive messages even in transmission mode and when messages are transmitted by using a broadcast communication medium. The problem can be avoided by sending synchronization messages with a random offset, even transmitting the particular offset with the message. Afterwards, the receiver can instantly rebuild the intended synchronization and achieve clock adjustment with respect to the received offset costs.

4.1. Deafness on MAC

We consider Figures 4 and 5 to explain the effect of deafness on MAC. For example, node N intends to transmit a data packet to node S. Node A is unaware to the ongoing communication between N and S. Therefore, MAC needs N to beamform in the direction of S and detect if the channel is idle for an (distributed coordinator function) interframe spacing DIFS duration. If the channel is idle, then N proceeds to the backoff phase and counts down the backoff counter while it is still beamformed toward S. While N is counting down its backoff counter, node S may intend to communicate with N. If A completes its own backoff before N and transmits an RTS to N, then N would not receive the RTS. In the absence of a reply from N, A would back off frequently and re-transmit the RTS until the dialog between N and S is over. Undeveloped re-transmission is a result of deafness. Consider a case in which N needs to transmit a packet to S. Once N has finished transmitting the first packet, it prepares to transmit the next packet by beamforming in the direction of S and then repeating the sequence of MAC processes, specifically, DRTS, backoff, DCTS, and carrier-sense, among others [59].

Unfairness is also a result of deafness. When several nodes attempt to communicate through node N, the node that wins channel contention retains the privilege to access the channel for a long time. Even though the receiver remains busy all the time, the transmitter nodes experience short-term unfairness [58].

4.2. Deafness in Firefly Clock Synchronization

Accurate clock synchronization in widely distributed systems is a difficult issue and challenging to accomplish. Nowadays, novel clock/time synchronization models have been developed based on the synchronization characteristics of fireflies. Early biological research was conducted by Richmond, who discovered the implicit mathematical synchronization model [61]. Fundamentally, firefly clock synchronization is based on pulse-coupled oscillators [62]. A simple model for synchronous firing of biological oscillators consists of a population of the same integrate-and-fire oscillators.

Tyrrell and Auer [63] discussed common applications of clock synchronization for WSN. A solution to
the deafness problem is the main outcome of the classification of clock synchronization. The deafness problem can be solved by classifying the synchronization cycle into two parts, first for local status updating and pulse firing, and second for eavesdropping on other firing sensor nodes. This can be easily accomplished by repeating the unique time T to 2T.

Wakamiya and Murata [64] presented clock synchronization based on data collection in WSN. Their design optimizes power management for periodic data collection in WSNs. In the illustrated approach, a base station-centric WSN consists of concentrically placed sensor nodes. Afterwards, firefly clock synchronization is used to distribute stimuli for the sensor nodes to calculate data and to transmit the result to the base station. Therefore, fully self-organized coordinated sensing can be accomplished. Babaoglu et al. [65] proposed a similar application synchronization scheme in overlay networks. Firefly clock synchronization has been applied as a robust and scalable heartbeat synchronization to address the synchronization issue in peer-to-peer networks, which is caused by network failures, scale, and dynamics.

4.3. Comparison among Existing Papers on Deafness Problem in a WSN

Observations of natural phenomena are considered the best source of information on spontaneous synchronization because natural phenomena closely match wireless sensor network response. The papers compared in Table 7, which address the issue of clock synchronization in WSNs, can greatly help ensure proper sensor network use and reduce energy consumption, thereby potentially increasing the lifespan of a sensor network.

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Methods</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>QoS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chodhury et al. (2002) [66]</td>
<td>This study considers two medium access control (MAC) protocols for an ad hoc network using directional antennas.</td>
<td>This study proposes multi-hop request-to-send (RTS) MAC (MMAC), which covers multi-hop RTS to gain the advantage of upper antenna gain over directional antennas that outperform 802.11. The issues of directional MAC and deafness are discussed.</td>
<td>This study neglects the design of more efficient directional MAC protocols and the impact of directional antennas on the performance of routing and other higher-layer protocols. No solution is offered.</td>
<td>End-to-end delay and throughput</td>
</tr>
<tr>
<td>Korakis et al. (2003) [67]</td>
<td>This study proposes a MAC protocol suitable for networks with directional antennas.</td>
<td>This study significantly reduces the hidden terminal and describes deafness problems.</td>
<td>The protocol does not assume any knowledge of the location of neighbors.</td>
<td>Throughput</td>
</tr>
<tr>
<td>Choudhury and Vaidya (2004) [58]</td>
<td>This study addresses deafness, a result of exploiting the beam-forming capabilities of directional antennas, and proposes a tone based on directional MAC protocol (ToneDMAC).</td>
<td>This study proposes ToneDMAC, which conserves the benefits of beam-forming while mitigating the adverse impacts of deafness on MAC layer performance.</td>
<td>This study does not clarify whether modifying 802.11 optimizes performance. MAC protocols designed specifically for directional antennas may be more efficient. The study also fails to explore the possibilities of using tones more effectively and analyze the effects of fading and interference. Thus, this is clearly false with regard to deafness, as discussed in this study.</td>
<td>Packet drops, end-to-end delay, and throughput</td>
</tr>
<tr>
<td>4. Takata et al. (2004) [68]</td>
<td>This study proposes a smart antenna-based wider-range access MAC protocol (SWAMP) for a wireless ad hoc network or smart antennas based on the IEEE 802.11 MAC protocol, which ensures spatial reuse and range extension through two kinds of access modes. The study shows that SWAMP is more effective with a proper load in multi-hop ad hoc networks. Qualitative evaluation against objectives is conducted.</td>
<td>This study addresses deafness. However, deafness was not solved by the proposed protocols.</td>
<td>Overhead, end-to-end delay, and throughput</td>
<td></td>
</tr>
<tr>
<td>5. Takata et al. (2006) [69]</td>
<td>This study proposes receiver-initiated directional MAC (RI-DMAC) to address deafness in directional MAC protocols for wireless ad hoc networks using a polling scheme. This study addresses deafness in directional MAC protocols. Potential deafness can be identified when the future receiver becomes idle and a packet is directly delivered after receiving ready-to-receive. Among the polling table nodes that may encounter deafness, the least recently transmitted node is used because a polled node yields fairness. The study also shows that RI-DMAC outperforms other directional MAC protocols in terms of throughput and fairness.</td>
<td>This study does not enhance RI-DMAC to incorporate quality of service requirements.</td>
<td>Fairness, end-to-end delay, and throughput</td>
<td></td>
</tr>
<tr>
<td>6. Jain and Agrawal (2006) [70]</td>
<td>This study proposes an algorithm for mitigating deafness in beam-forming antennas. The algorithm is implemented for two medium access protocols (MMAC with node-based backoff [MMAC-NB] and explicit synchronization via intelligent feedback [ESIF]) for multiple beam antennas. The performance of MMAC-NB improves, and ESIF implementation is simplified by this algorithm. The study shows that performance gains can be increased by the proposed algorithm.</td>
<td>The study develops an algorithm for mitigating deafness but disregards fairness and energy wastage.</td>
<td>End-to-end delay and throughput</td>
<td></td>
</tr>
<tr>
<td>7. Tyrrell and Auer (2007) [63]</td>
<td>This study proposes the modification of reference nodes where a common time scale can be imposed on a set of distributed oscillators by firefly synchronization. The study proposes solving deafness by modifying reference nodes by firefly synchronization. Deafness can be solved by dividing the synchronization cycle into two parts: (1) for local phase update and pulse firing and (2) for listening to other firing nodes. This division can easily be accomplished by repeating unique time (T to 2T).</td>
<td>The study proposes the modification of reference nodes but not normal nodes, which follow only a self-organized synchronization strategy. The nodes were modified only in the mesh network and not in wireless sensor networks (WSNs).</td>
<td>End-to-end delay and throughput</td>
<td></td>
</tr>
<tr>
<td>Reference</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. Korakis et al. (2008) [71]</td>
<td>This study proposes a MAC protocol to exploit directional antennas in wireless networks completely. This study reduces hidden terminals and deafness, which are the primary causes of decreased efficiency in directional transmissions in ad hoc networks. This study assumes no knowledge of the location of neighbors. Given the dynamic nature of protocol functionality, the protocol behaves efficiently in an environment with both static and mobile users.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. Jain et al. (2008) [60]</td>
<td>This study includes identifying fundamental issues related to medium access with multiple beam antennas by using a single channel, designing a cross-layer hybrid MAC that maximizes the benefits of multiple-beam smart antennas, and proposing wireless mesh network architecture with heterogeneous antenna technologies. This study argues that deafness cannot be entirely determined because no assurance exists that all nodes are informed of all incoming or continuing transmissions in its area when distributing to a single channel. Deafness can be mitigated only to a limited extent by taking either reactive or proactive actions. This study does not model the multipath effect, which occurs when multiple copies of the same signal are received by the receiver from different directions.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. Karapistoli et al. (2009) [7]</td>
<td>This study proposes a directional ultra-wideband MAC protocol (DU-MAC) that effectively addresses deafness and resolves the location of neighbors. This study proposes a protocol that outperforms the IEEE 802.15.4a omni mode standard in terms of throughput and energy consumption. Thus, jointly utilizing UWB transmission with directional communication in shared wireless medium sensor networks would be beneficial. This study does not examine the DU-MAC protocol in greater depth, that is, by finding a prediction mechanism based on a probabilistic model. In the study scenario, beam hops are probability-dependent and thus severely reduce the number of times preamble trailers are sent, and subsequently minimize the rotation phase before packet sending.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11. Ekbatanifard and Monsefi (2012) [59]</td>
<td>This study surveys and classifies state-of-the-art multi-channel MAC protocols proposed for WSNs. Deafness indicates that a receiver may be missing from the protocol. The synchronization column shows whether the protocol assumes that clock synchronization is externally required. The channel switching column lists the number of rate switching instances requested by the protocol in all steps.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This study suggests a multichannel protocol for high-bandwidth WSNs based on a combination of time-division and frequency-division multiple access to avoid collisions and the deafness scheduling of transmissions by using reinforced learning for joint scheduling and routing in each node. They are achieved more easily and more often, latency is lower, and packet loss is smaller. The proposed protocol also shows better performance in terms of end-to-end delivery rate, end-to-end latency, and high energy efficiency. The authors aim to improve network throughput by scheduling node transmissions to avoid collisions and deafness in multichannels. However, the study provides only collision-free operation. This study also does not implement the proposed protocol in combination with an appropriate time synchronization protocol on an operational sensor node test bed nor evaluate its performance with real devices.

This table presents studies that address deafness from different perspectives. Studies 1, 2, 3, 4, 6, 8, and 9 adopt the antenna perspective to avoid deafness. Studies 5, 7, 10, and 12 take the MAC protocol perspective to mitigate deafness.

5. Conclusions and Future Research Directions

Observations of natural phenomena are considered the best source of information on spontaneous synchronization. Natural phenomena tend to match wireless sensor network responses closely. A comparison among papers, presented in Table 7, can facilitate the process of ensuring clock synchronization in WSNs. In addition, this study can significantly help not only in ensuring efficient sensor network use but also in reducing energy consumption and, therefore, increasing the potential for increasing the lifespan of a sensor network. This work is a useful source on clock synchronization that can provide WSN researchers and novices with a better understanding of the deafness problem to promote effective designs and systems.

6. References


