IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 15, 2020, accepted August 12, 2020, date of publication August 25, 2020, date of current version September 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3019392

Median Access Control Protocols for Sensor Data Collection: A Review

WENBO ZHAO[1], (Member, IEEE), YIFAN LI[2], BO YAN[1], AND LUPING XU[1]
[1]School of Aerospace Science and Technology, Xidian University, Xi'an 710071, China
[2]School of Logistics Engineering, Wuhan University of Technology, Wuhan 430070, China

Corresponding author: Wenbo Zhao (wbzhao@xidian.edu.cn)

This work was supported in part by the National Science Foundation of China under Grant 61502373, and in part by the Natural Science Foundation of Shaanxi Province under Grant 2020JM-187.

ABSTRACT Median Access Control (MAC) protocols are designed to mitigate collisions and enhance the energy efficiency for sensor data collection. This paper reviews two basic categories of MAC protocols. The first class is the contention-based protocols, where nodes randomly compete for channel access. The second class is the schedule-based MAC protocols, in which nodes access the channel on the basis of the predetermined schedules. We focus on discussing the Time Division Multiple Access (TDMA) protocols and classify different TDMA schedulings into three categories according to the communication patterns in the network, i.e., the link scheduling, aggregate scheduling and non-aggregate scheduling. Link scheduling deals with the peer to peer communication pattern, where there is no central node in the network. In comparison, both the aggregate and non-aggregate schedulings handle the convergecast communication pattern, in which all traffic are destined to the sink. This survey provides a comprehensive overview of these three categories and provided a detailed briefing on how the TDMA schedules handle the network traffic dynamics in the network. Compared with other surveys in this domain, this review does not confine itself to deal with collecting a particular form of data, but provides a unified framework to integrate the data semantics in a broader sense into the design of TDMA scheduling algorithms.

INDEX TERMS TDMA scheduling, sensor data collection, wireless sensor networks, survey.

I. INTRODUCTION

Recent technological development in micro-electronics, communications and micro-processor systems has made possible the low manufacturing price for small wireless sensor devices (sensor nodes), which are adept at sensing and measuring various phenomenons in physical world. Large-scale deployment of sensor nodes forms wireless sensor networks [1]. Such networks are often deployed in the targeted area to continuously surveil changes in the environment.

Sensor data gathering is a major process involved in many wireless sensor network applications such as environmental monitoring and target tracking [53], [56]. In these applications, sensor nodes generate data every sampling interval and they coordinately transmit the data to a centralized controller (base station or sink node). The base station is connected to the personal computer of a remote user for further analysis and process. Sensor nodes are highly constrained by system resources, e.g. energy power, computation ability and memory storage, while the base station is equipped with sufficient system resources.

To gather data in the network, one simple approach is to pull all the *raw data* acquired at the individual sensor nodes to the base station and then process them at the base station. This is the *raw data collection* or *non-aggregate data collection* by definition [1]. Raw data are the original observations of individual sensor nodes. Nevertheless, owning to the data-intensive nature of monitoring applications, the raw data collection poses heavy burden of communication on the underlying network.

By contrast, another approach gathers the *aggregate forms of data* by applying a process of combining sensor data from different sensor nodes, e.g., the maximum sensing value of all the sensor nodes. As data are usually aggregated at sensor nodes as the data flow through them, it is known as *in-network aggregation* [1], [35], which is more efficient in reducing the energy consumption at individual sensor nodes. Performing the in-network aggregation also contributes to enhancing the

The associate editor coordinating the review of this manuscript and approving it for publication was Alessandro Pozzebon.

network security since eavesdroppers get less chance to intercept messages due to the reduced workload in the network.

A. CHALLENGES OF DATA GATHERING

Sensor data gathering confront two main challenges. The first challenge is the communication interferences inherent in the wireless communication. To gather data generated in the network, sensor nodes communicate with one another and the base station through radios. Sensor nodes that communicate on the same frequency share the same wireless communication channel. If concurrent transmissions from multiple nodes are inappropriately handled, data may be scrambled at the receivers due to collisions. To collect sensor data efficiently, collisions must be reduced as much as possible.

Second, frequently performing radio operations will quickly deplete energy of sensor nodes and render the whole network useless. Sensor nodes are battery-powered equipment and the radio communication is the primary consumer of sensor batteries [1]. Thus, excessive radio communication should be avoided in the data collection to conserve sensor batteries.

Median Access Control (MAC) protocols are devised to reduce the effect of collisions as well as conserve sensor energy. Generally, as shown in Fig. 1(a), there are two basic categories of MAC protocols: contention-based MAC protocols [4]–[6], [8]–[17] and schedule-based MAC protocols [20]–[28], [31]–[33], [36]–[52], [54]–[62], [64]–[99]. In contention-based MAC protocols, nodes randomly access the channel, while in schedule-based MAC protocols, nodes get access to the channel based on predetermined schedules.

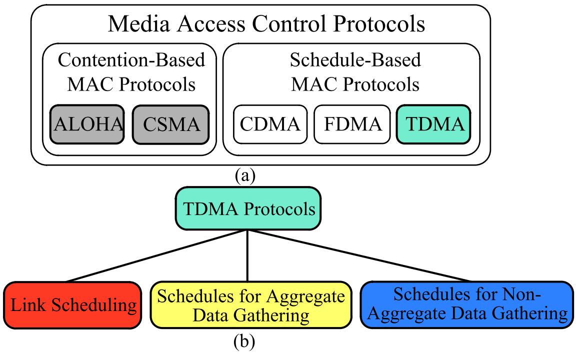

Media Access Control Protocols
Contention-Based MAC Protocols
ALOHA
CSMA
Schedule-Based MAC Protocols
CDMA
FDMA
TDMA
(a)
TDMA Protocols
Link Scheduling
Schedules for Aggregate Data Gathering
Schedules for Non-Aggregate Data Gathering
(b)

FIGURE 1. The structure of this paper: (a) Classification of MAC protocols. (b) Categories of TDMA scheduling protocols.

TDMA, FDMA and CDMA (Time/Frequency/Code Division Multiple Access) are three commonly used and contention-free channel access methods that build transmission schedules to eliminate collisions. FDMA requires implementing multiple radio channels in the network, which increases the hardware complexity of sensor nodes [8]. CDMA requires each node to perform complex encoding and modulating, which incurs the high computation cost and defeats the goal of energy efficiency. In comparison, TDMA brings no additional hardware requirement and incurs less computation cost. In wireless sensor networks, TDMA is the most attractive one for its simplicity.

TDMA divides the continuous time into discrete slots and eliminates interferences by arranging only non-conflicting transmissions to carry out in the same time slot [3]. Compared to contention-based MAC protocols, TDMA circumvents the energy expense and latency overhead required to seize the channel and to conduct retransmissions in case of collisions. Additionally, sensor nodes are able to shut down their radios whenever they do not need to send or receive any data packet, further conserving energy at sensor nodes.

B. RESEARCH MOTIVATION

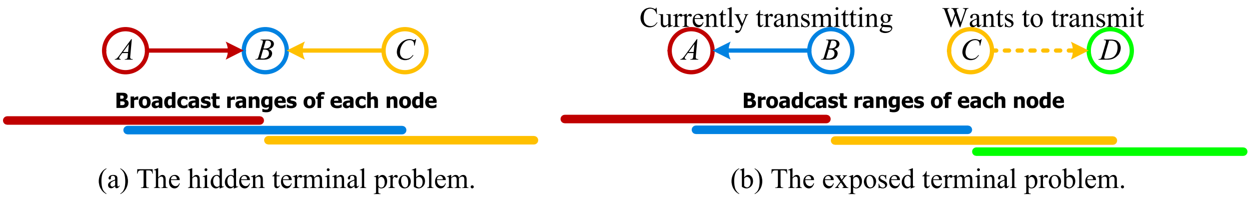
This survey investigates both the contention-based and schedule-based MAC protocols. We focus on elaborating various TDMA scheduling techniques in sensor data collection. TDMA scheduling can be classified from many different perspectives [3]: link scheduling and node scheduling protocols, topology-transparent and topology-dependent protocols, centralized and distributed protocols, etc.

It should be noted that although there exists many surveys in this domain, these efforts are made from a more nuanced perspective. They focus on summarizing the using of TDMA in some dedicated networks to collect a particular form of data, e.g., such as schedulings in wireless multi-hop networks [3], link scheduling for underwater acoustic sensor networks [34], or aggregate scheduling in sensor networks [35]. *By contrast, this paper does not confine itself to a particular form of data, but provides a unified framework to integrate the data semantics in a broader sense into the design of TDMA scheduling algorithms.*

In this survey, we classify various TDMA schedules from a new perspective based on *different communication patterns* in the network. The communication pattern indicates how the data flow transfers among sensor nodes in the network, i.e., who communicates with whom and the number of transmissions that each node participates in.

According to this, these scheduling algorithms mainly fall into three categories (as shown in Fig. 1(b)): *link scheduling*, *aggregate scheduling* and *non-aggregate scheduling*.

- Link scheduling deals with the *peer to peer* communication pattern, where there is no central node in the network. One time slot is assigned to each link to enable each node communicate once with each of its neighbor.
- In comparison, both the aggregate and non-aggregate schedulings handle the *convergecast* communication pattern, in which all traffic are destined to the sink. In aggregate scheduling, each node is assigned only one transmission slot. This is because multiple packets received downstream can be combined to form only one partial aggregate result, which is then packed up into one packet for delivery. To facilitate the in-network aggregation, strict limits on the relative transmission order between a parent and its children must be obeyed such that the sending slots of a parent node always appear after those of all of its child nodes in the schedule.



(a) The hidden terminal problem.

(b) The exposed terminal problem.

FIGURE 2. The hidden and exposed terminal problems in the CSMA/CA mechanism (taken from [2]).

- There is no such requirement to decide who always has the priority to transmit first beween a parent and its child nodes for non-aggregate scheduling. Furthermore, in non-aggregate scheduling, an internal node is assigned a number of sending slots to forward its local packet, as well as intactly relay packets generated by its descendants in the routing tree.

On the other hand, computing and broadcasting schedules both require messages being exchanged among different nodes in the network, which brings about extra time and energy cost. Thus, the schedule constructed is expected to be utilized for data gathering for as long as possible. However, a transmission schedule is usually pre-calculated corresponding to a certain kind of workload in the network, and schedules under different workloads possess totally different arrangement of transmissions. In practical data gathering, the workload often dynamically changes over time in an unpredictable manner. To avoid constructing and deploying new schedules repeatedly from scratch, it is desirable to make the schedule adapt to the traffic dynamics in the network. We have also addressed this issue in the review.

The *contributions* can be briefly summarized as follows:

- In this paper, both the contention-based and schedule-based MAC protocols are surveyed. Our focus is studying various TDMA scheduling algorithms.
- Based on different network communication patterns, we classify TDMA scheduling algorithms into three categories: link scheduling, aggregate scheduling and non-aggregate scheduling, each of which is elaborated in a separate section.
- We provide discussions on how the TDMA schedules handle the traffic dynamics in the network.

The reminder of this paper is organized as follows. Section II summarizes the contention-based MAC protocols, and the schedule-based MAC protocols are comprehensively elaborated in Section III. Section IV provided detailed briefings on how the TDMA protocol handles the traffic dynamics. We provide a brief discussion on future works in Section V, and Section VI summarizes the survey.

II. CONTENTION-BASED MAC PROTOCOLS

The ALOHA protocol [4] and Carrier Sense Multiple Access (CSMA) [5], [6] protocol are the first two median access protocols used in the DARPA Packet Radio Network (PRNET) [7], which is the first ad hoc multi-hop wireless network. ALOHA protocols adopt the best effort service by assuming a clear channel before transmissions and retransmitting data upon collisions. Specifically, pure ALOHA allows a node (or a station) to transmit at any time. Slotted ALOHA divides the time into discrete time slots and transmits data only at the beginning of each slot. In general, ALOHA protocols work well under the low workload scenario. When the workload becomes heavy, high cost is paid for the frequent data retransmissions.

CSMA protocols [5], [6], on the other hand, outperform ALOHA in the heavy workload scenario and gradually become popular for its easy implementation and good scalability. CSMA protocols reduce collisions by enabling each node to verify that the channel is idle before sending data. To achieve this, a node listens for the possible carrier wave in the channel. If the carrier wave is detected, the sensor node backs off for some time, then senses the channel again for transmission. Otherwise, the node sends data immediately. CSMA with collision avoidance (CSMA/CA) further reduces collisions among neighboring nodes [5], [6]. In CSMA/CA, once a node detects an idle channel, the node broadcasts a signal to its neighbors telling them not to send data during its own transmission. However, CSMA and CSMA/CA do not eliminate collisions completely due to *the hidden terminal* and *the exposed terminal problems*.

By definition the hidden terminal problem denotes the collision caused by concurrent transmissions from nodes that are not directly connected with each other. Consider the configuration depicted in Fig. 2 (a), where node B can communicate with both nodes A and C, but A and C cannot directly hear each other. Suppose node A is now delivering a packet to node B. When C wants to send a packet to the node B, C listens to the channel and mistakenly judges that the channel is currently available. C performs transmission and the collision happens at B. This is because C is hidden from A and thus cannot detect A's transmission signal. The exposed terminal problem, on the other hand, refers to the unnecessary deferring of transmissions that could have happened at the same time with the current transmission. Consider the example shown in Fig. 2 (b), where node B is delivering data to node A. If node C wants to send a packet to D, node C senses the channel and it detects the carrier of B. Thus, C defers its data transmission. However, it is not reasonable to postpone the transmission of C since the other receiver A is located beyond the communication scope of C.

To solve the hidden terminal problem, the distributed coordinate function (DCF) can be used to supplement

CSMA/CA [5], [6]. Specifically, DCF works by exchanging a packet of Require To Send (RTS) as well as a packet of Clear To Send (CTS) in the network. Sending nodes who detect an idle channel further contend the channel by transmitting RTS packets to receivers. A receiver answers to the first RTS sender with a CTS packet notifying the sender to perform data transmission. Other nodes that receive either RTS, CTS, or both have to keep quiet for a given time, which is predefined in the RTS and CTS packets. As pointed out by Karn [8], the use of RTS and CTS packets also relieves the exposed terminal problem. If a node A hears the RTS packet destined to another node B, A continues sensing the channel for a given time. If A does not hear the corresponding CTS packet from B, it indicates B is beyond the communication scope of A. Thus, A could transmit without fear of interfering the data receiving of B.

The above basic CSMA protocols are originally designed to reduce collisions in a single-hop network, and require nodes to keep listening all the time for possible transmissions. If these protocols are directly applied to a multi-hop sensor network, a lot of sensor energy would be wasted on idle listening, overhearing and transmitting control packets (RTS and CTS packets). Idle listening refers to the scenario that a sensor node keeps sensing the wireless channel but receives nothing. Overhearing denotes the scenario that a node continuously listens to transmissions of packets destined to other nodes. MAC protocols tailored to wireless sensor networks improve the basic CSMA mainly on reducing the energy expensed at idle listening as well as overhearing. MAC protocols tailored to wireless sensor networks could be broadly divided into two categories: *synchronized protocols* (S-MAC [9] and T-MAC [10]) and *unsynchronized protocols* (W-MAC [11], B-MAC [12] and X-MAC [13]).

A. SYNCHRONIZED PROTOCOLS

Ye *et al.* [9] proposed the S-MAC (Sensor-MAC) protocol. Three key techniques are involved in the S-MAC protocol: periodical sleeping, virtual clustering and adaptive listening. In S-MAC, nodes periodically go to sleep and become active. As shown in Fig. 3, the time is partitioned into discrete frames. Every frame is constituted by a sleep interval and an active interval. During the sleep interval, nodes turn down their radio to preserve energy. During the active interval, nodes communicate with neighbors to transmit the packets queued in the sleep period. At the very beginning of the active interval, the information of synchronization and schedule is exchanged among neighboring nodes to make sure that they wake up together next time. Hence, nodes in the network naturally form virtual clusters. Sensor nodes in each cluster get synchronized and adhere to the same schedule. Nodes on the boundary of two virtual clusters comply with the schedules of both clusters to guarantee the connectivity of the network. The virtual clustering technique reduces the dependence on the system-wide time synchronization. Following the synchronization phase, transmissions are performed following the RTS-CTS-DATA-ACK mechanism [5], [6], and both the sender and the receiver ignore the sleep schedule until they finish transmitting the data. The same authors also proposed an adaptive listening method to avoid overhearing and bring down the latency of data gathering [14]. Once a node detects an RTS or CTS destined to its neighbor, it turns to the sleep state to avoid overhearing the transmission of the data packet. The node wakes up briefly at the end of the transmission to sense the channel. If the node is just the next hop on the route of data delivering, this briefly waking up can reduce the latency since data can be immediately forwarded to this node without waiting until its next scheduled active period.

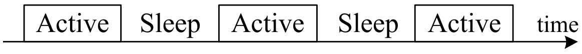


FIGURE 3. Periodic listen and sleep (taken form [9]).

In S-MAC, the lengths of the active and sleep period in each frame are fixed. No matter there is actual traffic or not, a node has to keep its radio on from the beginning of an active period to the end. This strategy is effective when the traffic is heavy, since the radio is fully utilized. However, the traffic in the wireless sensor network usually fluctuates over time. Setting a long active period results in a lot of idle listening when the workload is light. On the other hand, setting a short active period leads to quite a long latency when the workload is heavy. To improve this, Dam and Langendoen [10] proposed the T-MAC protocol that further mitigates idle listening by making the length of the active period adapt to the workload. T-MAC utilizes a short time-out window to control the length of the active time period. A sensor node can choose to turn to sleep before the end of the active interval, if it does not receive or transmit any data in a time-out window. Simulation results indicate that T-MAC outperforms S-MAC in achieving less energy expenditure of sensor nodes. Because of the early sleeping of sensor nodes, however, T-MAC sacrifices latency for the reduced energy consumption.

In general, both the S-MAC and T-MAC protocols endure the scaling problem. With the increase of the network scale, more synchronization information is exchanged in the network and more schedules have to be maintained at each node for its neighbors. In addition, these protocols lead to imbalanced energy expenditure of different nodes in the network. Because nodes on the boundary of the two virtual clusters comply with both schedules to assure the connectivity, they get less opportunities to sleep and spend more energy. These drawbacks motivate the designing of unsynchronized MAC protocols (W-MAC [11], B-MAC [12] and X-MAC [13]), which adopt the Low Power Listening (LPL) [15] technique to facilitate on-demand data transmissions. The basic idea of LPL is presented in Fig. 4. The sender uses an extended preamble as an advanced notice of the packet. The receiver periodically wakes up to probe the channel. If a preamble is detected, the receiver keeps awake until it completes receiving the packet. If no preamble is discovered, the receiver

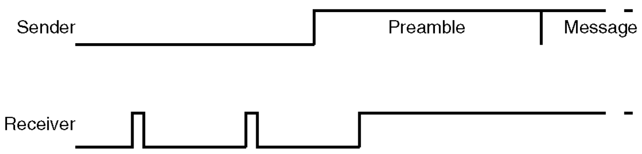


FIGURE 4. Low-power listening (taken form [15]).

turns to the sleep state immediately. By transmitting the extended preamble, LPL shifts the energy consumption from the receiver to the sender, thereby reducing idle listening.

B. UNSYNCHRONIZED PROTOCOLS

W-MAC proposed by El-Hoiydi and Decotignie [11] is designed to facilitate efficient data dissemination from the sink to all the sensor nodes. The authors assumed that the base station has enough transmission power to communicate with the farthest sensor node in the network. Sensor nodes are required to piggyback their next active times in the acknowledgement packets to the base station. The base station then schedules its transmission slightly earlier than the next active time of a node. W-MAC devises a mechanism to make the length of the preamble adapt to the packet inter-arrival time. Intuitively, the shorter the packet inter-arrival time, the shorter the preamble length. By doing this, W-MAC naturally mitigates the overhearing in the heavy traffic scenario, since the short preamble makes it less possible to be detected by overhearers. In the light traffic scenario, it is possible that the preamble is much longer than the data packet. In this case, W-MAC includes and repeats the data packet for several times in the preamble. A node only needs to receive and analyze the first data packet contained in a preamble to decide whether to keep active till the end of the data transmission and send an acknowledgement packet to the base station. Therefore, an overhearing node can go to sleep earlier on finding that it is not the targeted receiver.

W-MAC is effective in achieving low power communication. However, it does not address how to reconfigure the period that nodes sense the channel when the workload changes. The B-MAC protocol designed by Polastre *et al.* [12] supports the adaptive sleep scheduling to meet the demand of the dynamic workload. Unlike S-MAC and T-MAC, which require the network layer support (such as the RTS-CTS mechanism for channel arbitration and message fragmentation in transferring bulks of data), B-MAC is a pure link layer protocol that contains only a small number of core media access functionalities. B-MAC provides the network layer supporting by developing a set of software interfaces, specifically, CCA (clear channel assessment), LPL (low power listening) and acknowledgement. It allows applications to control and reconfigure these interfaces based on the current workload. By factoring out higher layer functionality, B-MAC has a small code size and requires less RAM space. B-MAC performs better than many existing protocols in aspects of latency, throughput, fairness and energy consumption for most instances.

X-MAC proposed by Buettner *et al.* [13] further saves energy by letting a sender transmit several short preambles instead of one long preamble before sending a packet. Each short preamble includes the identifier of the receiving node. A sender pauses to listen to the channel between transmitting two successive preambles. Nodes in the network probe the channel periodically. On detecting a short preamble, a node checks whether it is the targeted receiver. If so, the node feedbacks an acknowledgement to the sender in the short pause, notifying the sender to stop transmitting any more preamble and immediately transmit data packets. Otherwise, the node switches to the sleep state to save energy. X-MAC also supports the adaptive listening technique to reduce the latency. It requires each receiver to keep sensing the channel for some time after receiving data. If a sender S who intends to send data to a receiver R overhears an acknowledgement from R to a node other than S, S backs off for a given time to avoid possible collisions from multiple senders. Then, S directly transmits data to R without attaching any preamble, because S can make sure that R is still probing the channel. The duration of the back-off time is long enough to enable the initial sender to finish the data transmission meanwhile short enough to ensure the targeted receiver is still listening to the channel.

Besides the protocol design works, other studies focused on analyzing the performance of the CSMA-type protocols. Laufer and Kleinrock [16] have investigated the capacity performance of CSMA/CA protocol under the constraints that 1) each node freezes its packet arrival process during a back-off period, and 2) the buffer of all nodes are not saturated. They showed that the stability and throughput results of CSMA/CA networks are predictable. However, the authors did not provide results for even the smallest unsaturated CSMA/CA networks. In fact, it is very difficult to analyze such models since the traffic dynamics in the network would affect the nodes' behavior in very intricate ways. Moreover, their assumptions are too stringent to be applied to practical systems. Wang *et al.* [17] have developed stochastic geometry models to analyze the mean throughput gains due to full-duplex transmissions in a multi-cell CSMA wireless network. They have studied the throughput gain under different link distance, interference ranges, network densities as well as carrier sensing schemes.

In summary, CSMA protocols have the advantages of easy to deploy and well adapting to the changing network topology. However, it cannot guarantee a bounded latency for data collection. As the network traffic becomes heavy, the probability of collisions of data packets and control packets increases, resulting in extra time spent on retransmitting the corrupted packets. In practice, however, many sensor applications require data to be delivered to the base station as soon as possible for timely processing [18]. This motivates the design of the schedule-based MAC protocols that provide guaranteed delay in data collection.

III. SCHEDULE-BASED MAC PROTOCOLS

TDMA, FDMA and CDMA are three major channel access methods for eliminating collisions. FDMA allocates one or more frequency channels to each node. It requires the use of multiple radio channels in the network and the implementation of precise filtering on the radio frequency to minimize the interference between adjacent channels [19]. These requirements would increase the hardware complexity of sensor nodes. CDMA requires each node to perform complex encoding and modulating, which introduces high computation cost and may defeat the purpose of energy conservation. TDMA is a commonly used channel access method that builds transmission schedules to eliminate collisions. Only non-conflicting transmissions are allowed to be scheduled in the same time slot. TDMA has no additional hardware requirement. Building TDMA schedules for sensor data collection in wireless sensor networks has been broadly researched in recent years.

In this section, we focus on reviewing various TDMA scheduling algorithms for sensor data collection. We classify various TDMA schedules based on different communication patterns in the network. The communication patterns show how the data flow transfers within a group of sensor nodes. It stipulates the communication objects in the network and the number of transmissions that each node needs to accomplish.

It should be noted that the implementation of TDMA protocols requires nodes to be synchronized and adhere to the same transmission schedule, and thus it is often criticized for the computation complexity and time overhead involved in the synchronization process. In some sensor applications, however, time synchronization by itself is highly necessary due to the nature of the applications, to guarantee the right comprehension of spatial-temporal correlation among sensor data generated at different sensor nodes. In target tracking, for example, targets cannot be correctly traced unless reports of the azimuth angle and positions of targets are arrived on an accurate time sequence. Apply a synchronous MAC protocol in these applications will not incur any additional overhead.

Basis of Classification: According to this, these scheduling works mainly fall into three categories: *link scheduling*, *aggregate scheduling* and *non-aggregate scheduling*. Link scheduling is used to eliminate interferences in the point-to-point communications. In this case, there is no central node in the network, and each link is activated only once. By contrast, both the aggregate and non-aggregate schedulings form the all-to-one communication pattern. That is, all sensor nodes need to deliver data packets to the base station through multi-hop transmissions. The difference between these two lies in the different forms of data that the scheduling algorithms deal with. The aggregate scheduling is used to collect the aggregated forms of data, while the non-aggregate scheduling is applied to gather the raw data without any in-network processing. In aggregate scheduling, each internal node first computes a single piece of data by aggregating its local sensing value with the data received from all of its children. Then, the internal node sends the aggregation result upstream. To do this, the transmissions of each parent must be allocated after the transmissions of its children. In raw data gathering, however, an intermediate node has to forward all the data received from the child nodes as well as its local sensing value. There is no constraint on the order of transmissions between those of a parent node and those of its children.

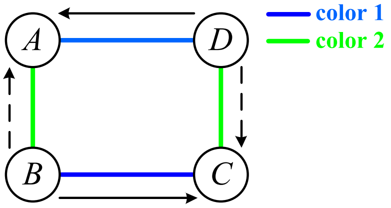


FIGURE 5. TDMA schedule based on the edge coloring (taken from [21]).

It should be noted that there also exists the one-to-all communication pattern in sensor data collection, when messages need to be broadcasted from the center (base station) to all nodes distributed in the network. Since the base station are equipped with plenty of energy, it can always increase its transmission power as high as possible to make every sensor node successfully receive the broadcasting message. Hence, the broadcast scheduling is not discussed in this review.

A. LINK SCHEDULING

Given a set of communication links, each having a unit traffic demand, the link scheduling mechanisms aim to build schedules for every sensor node to communicate with each neighbor for one time [20]–[28], [31]–[33].

Features of Link Scheduling:

- It serves for the peer to peer communication pattern, where there is no central node in the network.
- In these schedules, each communication link in the network is assigned only one transmission time slot.

According to the goal of optimization, we characterize the link scheduling into three major classes: shortest link scheduling, maximum link scheduling, and minimum age link scheduling.

1) SHORTEST LINK SCHEDULING (SLS)

Some works [20]–[22] aim to construct the minimum length schedules. However, finding the minimum number of time slots for link scheduling is proved to be an NP-hard problem [20]. Ramanathan [20] proposed a centralized heuristic algorithm called UxDMA, a unified framework for (T/F/C)DMA scheduling. The latency bound provided by UxDMA is within $O(\theta)$ of the optimal length, where θ is the thickness of the graph, i.e. the minimum number of planar graphs into which a given graph can be partitioned. Implementing UxDMA in a wireless sensor network requires collecting the complete network topology at the base station and distributing the schedule to nodes in the network, which is not scalable.

Gandham *et al.* [21] proposed a distributed link scheduling algorithm. This study is based on the classical edge coloring problem. In this problem, each edge corresponds to a communication link between two nodes. The problem tries to assign the minimum number of colors to edges such that no two edges indent on a node are dyed with the same color. A valid edge coloring for an undirected graph can be derived by utilizing up to $\Delta + 1$ colors [29], where Δ is the maximum node degree of the graph. However, directly mapping the edge coloring of an undirected graph to a TDMA scheduling may incur the hidden terminal problem. For example, a valid link coloring for a four node graph is shown in Fig. 5. The transmissions in time slot 1, according to the edge coloring, are B to C and D to A. The reception at nodes A and C are garbled due to collisions. To remedy this, the algorithm proposed in [21] works in two stages: in the first stage, a valid distributed edge coloring is computed; in the second stage, each color is mapped to a unique time slot and the direction of transmission along each edge is identified such that the hidden terminal problem is avoided. The constructed schedule contains at most $2(\Delta + 1)$ slots when the network topology is acyclic, where Δ is the maximum node degree.

Grandham's work is conducted on the basis of the assumption that the interference range of each node equals to the transmission range. In practice, this is not always true. Wang *et al.* [22] assumed that the transmission ranges and interference ranges of different nodes could be very different. They proposed conflict-free link scheduling algorithms with latency guarantee to maximize the network throughput. In these algorithms, each link is assigned the earliest possible time slot that does not incur interference with the already-scheduled links.

The above shortest link schedulings [20]–[22] are derived from the graph-based model in which the interference is treated as a pairwise constraint, i.e., a group of links are regarded as conflict-free if they are pairwise conflict-free. In actual wireless communications, however, the interference among concurrent transmissions is not pairwise but additive. The physical interference model is a more realistic and accurate model which uses the signal-to-interference-plus-noise-ratio (SINR) to depict the aggregate effect of interference in the network [30]. In this model, a transmission from node i to node j would be successful if and only if the received signal strength at j is at least the minimum SINR threshold required by node j. In general, the link scheduling under the physical interference model is a more complicated problem due to the additive interference among concurrent transmitting links in the network. Building the minimum-length schedule under the SINR-based models is proved to be NP-hard in [31].

2) MAXIMUM LINK SCHEDULING (MLS)

MLS targets at maximizing the number links that are scheduled simultaneously in one time slot.

a: CENTRALIZED SCHEDULING WITHOUT POWER CONTROL

Moscibroda *et al.* [32] have derived an upper bound on the schedule length under the exact SINR model. They assumed the transmit power of each node can be arbitrarily high, which is not realistic in practical use. Other works [31], [33] designed good approximation algorithms for MLS, by assuming nodes using the same constant transmission power. However, they have adopted an approximation of the SINR model that either does not consider the radio interferences from faraway transmitters [33], or totally neglects the effect of ambient noise [31]. Thus, the approximation bounds derived under these SINR-based models do not make sense in the exact SINR model.

Some other works consider a variation of MLS with throughput maximization under the SINR interference model [23]–[26]. Blough *et al.* [23] were the first to construct the minimum-length feasible schedules to optimize the overall network throughput, under the exact SINR model. They partitioned the network into a set of fixed length squares which are then four-colored to ensure any two adjacent squares are assigned different colors. Links whose receivers located in different squares with the same color can be activated concurrently without collisions. The authors have identified the ‘‘difficult to schedule’’ links that block the calculation of tight approximation bounds for this problem. They proved the deterministic approximation bounds on the schedule length when the number of such links is constrained by a constant.

Blough's work has three major disadvantages. First, their designs require global propagation of messages to make the scheduling decisions. Second, the approximation ratio of their algorithm is a linear function of the number of ‘‘difficult to schedule’’ links. In the general case, their works are lack of satisfactory theoretical guarantee. Third, the transmit power of each node is fixed to be the largest power that is adequate to support the transmission through the longest link, causing huge energy waste at short links and strong interferences in the whole network.

b: DISTRIBUTED SCHEDULING WITH POWER CONTROL

Zhou *et al.* [24] have tackled the challenges of Blough's work and proposed localized link scheduling algorithms for throughput maximization with transmit power control. The rationale behind their proposed method is that, the total interferences at a transmitting link could be effectively bounded, if such a link keeps enough distance away from all the other transmitting links. Their proposed method first divides the network into several disjoint local areas which are certain distance away from each other. Then, the local scheduling is conducted independently and concurrently within each local area. The partition of regions slightly changes at each time slot to ensure those backlogged links lying outside the local areas at a time also get chances to be scheduled at another. The proposed algorithms provided theoretical guarantee for MLS and kept the networks away from arbitrarily bad throughput performance.

The approximation factors achieved by Zhou *et al.* [24] is loose for two reasons. First, the side length which indicates the distance between two unit squares is fixed for all links in [24]. It fails to consider that the amount of interference that different links can tolerate is in direct proportion to its link length. Second, [24] fails to consider the possible concurrent transmissions such that one transmission is on a link in a local area and another exists in a region which does not belong to any local area.

To remedy this, Yu *et al.* [25] designed algorithms to conduct partitions on both the network area and the links. They divided the network into a set of hexagons which are then three-colored such that no two adjacent hexagons are given the same color. They further partitioned the links into non-overlapping subsets in a way that links in the same subset have roughly the same length. Meanwhile, the size of a hexagon is not fixed but is a function of the approximate length of links inside the hexagon. Compared with the method of partitioning the network into squares of the same sizes applied in [23], the hexagon partition can better make use of the parallelism of transmissions and is more effective in reducing the schedule length. In Yu's algorithm design, the ''difficult to schedule'' links are completely eliminated by increasing the sending power of sensor nodes by a little. Rigorous theoretical analysis has showed the approximation ratios of Yu's methods are tighter than the best known ratios.

c: SCHEDULING IN THE UNDERWATER ACOUSTIC SENSOR NETWORKS

The aforementioned scheduling mechanisms are designed for the terrestrial wireless networks. The underwater communication, on the other hand, transmits data through acoustic channel and poses great challenges to the MAC protocol design [34]. Bai *et al.* [26] have proposed link scheduling algorithms with power control for the underwater acoustic sensor networks. They have formulated finding the latency optimized and conflict-free link schedule as a Mixed Integer Linear Programming (MILP) problem, and devised a heuristic algorithm to solve the MILP. Simulation results showed their proposed method can increase the network throughput meanwhlie decrease the end-to-end latency.

d: SLEEP SCHEDULING

In all of the scheduling algorithms discussed above [20]–[26], links incident on the same node may be allocated non-consecutive transmission time slots and a sensor node may start up numerous times from the sleeping mode to the active mode for communicating with its neighbors. To avoid the extra latency and energy incurred by the frequent mode transitions, Ma *et al.* [27] proposed the contiguous scheduling algorithm that works by assigning consecutive transmission slots to incoming links (fan-in) associated with the same node. As a result, a sensor node in a tree network only needs to wake up two times in a sampling interval, i.e., one time for gathering all the data from its children and the other time for sending the data to its parent upstream in the tree. Wu *et al.* [28] proposed efficient scheduling methods to reduce the overheads of the mode transitions. They also devised an algorithm to build an energy conserving routing tree for data gathering.

B. TDMA SCHEDULES FOR AGGREGATE DATA GATHERING

Sensor networks are usually deployed in the network to gather both raw data and aggregate forms of data. Raw data are the original observations of individual sensor nodes. By contrast, aggregate forms of sensor data are the result of applying a process of combining individual sensor readings from different sensor nodes. In fact, data could be aggregated either at the base station, or at sensor nodes as the data flow through them. The latter is known as in-network aggregation, which is more efficient in reducing the number of transmitted packets and mitigating the interferences in the network. Aggregation can effectively reduce the risk of messages being exposed to and intercepted by eavesdroppers, and thus helps to improve the security performance of wireless networks. Schedulings for data aggregation have been reviewed in [35].

Features of Aggregate Scheduling:

- In this type of schedules normally each node is allocated only one time slot for transmission, since the partial aggregate results forwarded by different nodes to their parents have the same size.
- The transmission slot of an intermediate node is always arranged after the transmission time slots of its child nodes, since the in-network aggregation imposes a stringent precedence requirement on the transmissions of different nodes in the data collection tree.

It should be noted that in link scheduling, there is no such precedence requirement and it does not matter which link is activated first in the transmission schedule. In this section, we categorize different aggregate scheduling mechanisms [36]–[52], [54]–[62], [64]–[75] according to different optimization goals.

1) MINIMUM LATENCY AGGREGATION SCHEDULING (MLAS)

Much attention has been put to design minimum length schedules for aggregate data collection [36]–[52]. We classify these works by the interference models adopted.

a: SCHEDULING UNDER PROTOCOL-BASED INTERFERENCE MODELS

• *Separated Routing and Scheduling Phases:* Building the minimum-latency aggregation schedule in a multi-hop sensor network is proved to be an NP-hard problem [36]. Chen *et al.* [36] proposed a centralized heuristic algorithm that builds a schedule whose latency is bounded by $(\Delta - 1)R$, where Δ indicates the maximum node degree and R denotes the network radius, i.e. the hop distance from the sink to the farthest node in the network. Their schedule is constructed based on a shortest path tree. Since both Δ and R could be of the same order of the network size, the algorithm may result in high latency.

Huang *et al.* [37] came up a novel idea to reduce the latency bound. They built an aggregation tree on the basis of the maximum independent set and devised a centralized scheduling method based on the aggregation tree. The latency provided by their method is bounded by $23R + \Delta - 18$, where Δ contributes to an addictive factor rather than a multiplicative element. However, the schedule constructed by Huang *et al.* [37] is not conflict free, since the authors failed to consider all the possible collisions in assigning time slots [38]. To rectify this mistake, Yu *et al.* [38] devised a distributed scheduling mechanism that ensures the conflict-free data collection. Similar to Huang's work [37], this study exploits the maximum independent set technique for schedule construction, and takes into account all the possible collisions. The latency bound is $24D+6\Delta+16$, where D denotes the diameter of the network. D can be $2R$ in quantity (R is the network radius).

The studies above [36]–[38] have used the base station as the root. Xu *et al.* [39] devised a distributed scheduling method which chose the topology center of the region to be the tree root. By doing this the latency bound of aggregate data collection can be further reduced to $16R+\Delta-14$. In their proposed method, the root has to first gather the aggregated data generated by all nodes, then sends the aggregation results to the sink via the shortest route, which incurs an additional delay of at most the network radius R. This topology setting has been adopted by many later research works [41]–[43].

- *Routing and Scheduling Executed in Parallel:* In all of the approaches mentioned above, the tree construction and aggregation scheduling are performed in two consecutive and separated phases. The effect of the scheduling method largely depends on the routing structures formed. Later works simultaneously executed the aggregation scheduling and the tree construction processes [40]–[42]. Bagaa *et al.* [40] have recently devised a distributed aggregation scheduling method named *DICA* which further reduces the latency bound to $\left(\lfloor\frac{2\pi}{arccos(\frac{1}{1+\epsilon})}\rfloor+3\right)R+\Delta-4$, where $0.05 < \epsilon \leq 1$. Unlike other studies that first create a routing tree then schedule the network nodes based on the tree already formed, *DICA* needs no knowledge about potential parent nodes or child nodes during the scheduling process. It constructs the routing infrastructure and the transmission schedule concurrently in a distributed manner. The authors have designed a novel hardware framework to deal with implementation issues and validated the superiority of *DICA* within the framework using the real sensor test bed.

Inspired by Bagaa's work, Chen *et al.* [41] have addressed the problem of building the minimum latency schedule in duty-cycle sensor networks. In such networks, sensor nodes switch between the dormant status and the active status in a cyclic manner to save energy, and nodes can collect data only when they are active. If a sender intends to transmit a packet to some receiver who is not awake at the time, the packet would be buffered at the sender until the receiver becomes active again. The aggregation delay in duty-cycle networks is much influenced by the routing structures. This is because the number of active slots of sensor nodes in every working cycle is rather limited. If too many child nodes select the same node as parent in the routing tree, the actual transmissions of child nodes would be postponed dramatically. The authors [41] proposed a distributed algorithm called *DSAD* to solve the problem. Similar to *DICA* [40], *DSAD* constructs the latency optimized routing structures and transmission schedules simultaneously.

Li *et al.* [43] proposed a distributed scheduling method on a cluster-based constructed tree called $Clu-DDAS$, which is declared to provide an upper bound of $4R+2\Delta-2$ on delay. Later, this upper bound developed by Li *et al.* [43] is proved not correct [42]. Yousefi *et al.* [42] made amends and invented an efficient distributed scheduling method named FAST with the latency bounded by $12R+\Delta-2$ time slots. The key design feature of their work building the aggregation tree and performing the scheduling simultaneously. In the routing tree construction, they firstly used a Connected Dominated Set (CDS) of three-hops. In this CDS, the distance between any node pair in a maximum independent set is precisely three hops. The connected three-hop dominating sets outperforms the connected two-hop dominating sets employed in previous works of Yu *et al.* [38] and Xu *et al.* [39] in reducing the latency.

An assumption central to the works of [36]–[43] is that the interference range of each node equals to the transmission range. Wan *et al.* [44] devised a scheduling method that supplements the works of [36]–[43] in considering the scenario that the transmission range is smaller than the interference range. They proved that the latency bound of their schedule is within a constant factor of the shortest possible latency for aggregate data collection.

All works [36]–[44] discussed above assume that any data can be packed into one data packet and networks are organized as tree-like structures. Nguyen *et al.* [49] considered the scenario that an intermediate node may combine a number of packets received from its child nodes into one data packet for forwarding to its parent, whlie satisfying a limit α on the packet size. They proposed a novel non-tree based method. Clearly, when $\alpha = \infty$ the problem is exactly the same with other aggregation scheduling problems. When $\alpha = 1$, the problem is to build collision free schedules for raw data gathering without any in-network aggregation. Later, the authors incorporated the channel assignment techniques into their designs to further enhance the latency performance [50].

b: SCHEDULING UNDER THE PHYSICAL INTERFERENCE MODELS

All the above algorithms build upon the graph-based interference models, in which the interference relationship is pairwise and simple [22], [30]. The graph based models neglect both the accumulated interference in wireless communications and the influence from faraway nodes beyond a certain range. Several other algorithms [45]–[48] base their works upon the physical interference models, i.e., SINR models. Generally, it is harder to guarantee all the active links meet the SINR limits, since the SINR model does not deal with interferences of links individually but consider the aggregated influence of potential interference from faraway nodes.

- *Basic Framework of Constant Power Model:* Li *et al.* [45] have first considered the problem of minimum latency scheduling under the SINR model. To prevent long links with small path-gain from disturbing possible simultaneous transmissions, the authors used short links as strong connected links to form a reduced network of the original communication graph. They constructed a connected-dominating-set (CDS) tree and utilized it as the routing tree. In CDS tree, nodes are regarded as dominators or dominatees. A dominator aggregates all data from its dominatees in different time slots. Data are then forwarded toward the root through a path consisting of dominators only in the bottom-up manner. To avoid interferences, the grid partition and coloring method were applied to ensure any link pairs that transmit simultaneously be separated far enough. The authors designed a an approximation algorithm of constant factor. The algorithm yielded a latency bounded by $O(R + \Delta)$ time slots, where sensor nodes are uniformly randomly deployed. Δ and R are the maximum node degree and the graph radius respectively in the reduced network.
- *Extensions of Other Power Models:* Based on this basic framework [45], including the tree construction, grid partition and coloring, following researchers [47], [48] working under SINR models tried to reduce the latency by replacing the constant power assignment by other power models. An *et al.* [47] proved the NP-hardness of the *MLAS* problem under the SINR model, and they derived an $\Omega(log\, n)$ approximation lower bound. Under the assumption of the dual power model, they proposed an approximation algorithm whose latency is upper bounded by $O(R+\Delta)$. The algorithm assigns each node either the high power level or the low power level, according to the position of the node in a CDS tree. They also cut the network into grids of two different area sizes to aggregate data generated by nodes with different power levels.

One drawback of Li's work [45] is, their algorithm offers no performance guarantee when applied to arbitrary topologies, since sensor nodes are assumed to be equipped with the same and constant transmission power. Li *et al.* [48] assumed the transmit power of every sensor node is large enough to reach the farthest node in the network. By trading the energy efficiency for time efficiency, they designed a fully distributed algorithm bounded by $O(K)$ time slots for the arbitrary network topologies, where K is the link length diversity which is defined as the logarithm of the ratio between the lengths of the longest and shortest links in the network. In their work, the routing and scheduling are not separately and sequentially executed but are jointly conducted with the power control. Initially, the network is partitioned into small cells according to K. The scheduling algorithm first used short links of the low power to aggregate data in small grids, then the data collected from small cells is further aggregated using long links of the high power in large areas. The two iterations repeat until all data is finally aggregated to the root.

- *MLAS for Rechargable WSNs:* The aforementioned MLAS algorithms are designed for Battery-Powered WSNs (BP-WSN), where the latency is mainly due to the need for each transmission to wait for good opportunity to squeeze into the earliest possible transmission slot while avoid collisions. Recently, the galloping development of energy harvesting technologies brings about energy self-sustainable networks called Battery-Free WSNs (BF-WSNs), in which sensor nodes can capture energy from the environment. Chen *et al.* [51], [52] investigated the MLAS problem for BF-WSNs, where the main cause of latency is not conflict avoidance but attributes to the *recharging time of sensor nodes*. In [51], Chen *et al.* defined the collision caused by the battery level constraint as the energy-collision, i.e., a node intends to receive or send a data packet but fails to do so since it has not yet harvested enough energy. They proposed a tree construction algorithm and three scheduling algorithms, by comprehensively considering the residual energy, energy-collision and interference constraints.

The algorithms proposed in [51] are centralized ones and have two drawbacks. First, to obtain the schedule, the base station is supposed to collect the information of the current energy of sensor devices, perform the scheduling, and then disseminate the schedule computed back to nodes in the network. Since the energy conditions of sensor nodes are time-varying, the whole process needs to be frequently conducted. The overhead involved may overweigh the benefit of utilizing such algorithms. Furthermore, the latency can be extremely high since it is possible that the node with the lowest recharging rate has a large number of children.

To remedy these drawbacks, in [52] the same authors proposed distributed tree construction and scheduling algorithms which can effectively adapt to the varying energy conditions in the network. Instead of aggregating data from all nodes, a subset of nodes is dynamically picked out for aggregation on the basis of their recharge rates and residual energies, while satisfying the given requirement on the coverage quality. They proved by simulation results that their proposed methods perform better than the centralized algorithms and can efficiently reduce the energy and time overhead in the process of scheduling.

2) DEADLINE CONSTRAINT AGGREGATION SCHEDULING (DCAS)

Although the in-network data aggregation can effectively reduce the network traffic, it also imposes additional delay at each internal node to wait for gathering all the packets from its child nodes. In some real-time surveillance applications, users sensitive to the latency may not even tolerate the latency achieved by MLAS. In target tracking, for example, if the duration of data aggregation takes too long, the estimated location of a moving object may substantially deviate from its actual position [53]. On the other hand, incorporating the participation of only a portion of sensor nodes but not all of them into aggregation can further reduce the delay. However, it also reduces the Quality of Aggregation (QoA), i.e., the amount of information extracted from the network. Much research works [54]–[61] have been done on the Delay Constraint Aggregation Scheduling (DCAS), which aims at maximizing QoA under the constraints of interference and the maximum tolerable delay designated by applications. The core of DCAS is to wisely decide the set of nodes to participate in data aggregation and the order of their transmissions. DCAS is quite different from MLAS in both the optimization objective and limitations, thus requires designing completely new solutions.

a: BASIC FRAMEWORK

Hariharan and Shroff [54] proposed a general optimization framework to maximize the aggregated information for a given tree while respecting the application-specific deadline. They defined the aggregated information as the number of nodes whose packets have been accounted for at the base station within the designated deadline. For simplicity reasons, they adopted the one-hop interference model such that any two links sharing a same node cannot be activated concurrently. The authors first reduced the scheduling problem to the Maximum Weighted Matching (MWM) problem, then devised a Dynamic Programming (DP) based algorithm using only local information at each hop to make the best aggregation and scheduling policies, i.e., who participates in aggregation and the transmission times of the participating nodes. The proposed DP algorithm is proved to result in an optimal solution with polynomial time complexity, which is much lower than that of the traditional MWM problem, since the matching problem is solved within each single hop but not across the entire network.

b: CONSIDERING UNRELIABLE LINKS

Based on the basic optimization framework above, the same authors extended their works to deal with the unreliable communication links [55]–[57]. Errors across different links are assumed to be independently distributed. In [55], Hariharan *et al.* first explicitly took into consideration the unreliable links and formulated an integer optimization problem to maximize the QoA at the base station, subjected to constraints of the deadline and interferences. They found that the inclusion of link errors substantially increased the difficulty of solving the problem, and the integer programming is MAX SNP-Hard. They proposed a sub-optimal version of the problem, which was then solved by a low complexity, distributed solution. To achieve this, they made a vital assumption that for any internal node in the network, the transmission order of its child nodes is already known. In [56], the authors formulated an integer programming problem which explicitly accounted for unreliable links and per-node energy constraints. They developed a distributed solution with low time and message complexity, to wisely allocate the transmission and reception energies at every sensor node such that the QoA at the base station gets maximized.

Although neither the deadline nor the interference was considered in [56] and the authors did not build any transmission schedule, the algorithm developed in [56] served as a building block for their later work presented in [57], which considered a problem more general than problems considered in [55] and [56]. The work of [57] explicitly accounted for the per-node energy constraints, unreliable links, deadlines and interference. In [57], the authors formulated a combinatorial optimization problem and proved this problem was NP-har by reducing it from a 3-partition problem [62]. Based on the idea of dynamic programming, they proposed a polynomial-time optimal algorithm for the case that the maximum node degree k of the aggregation tree roughly equals to $O(log\, N)$, where N denotes the total number of sensor nodes. For a denser sensor network with a larger k value, they further looked at a suboptimal version of the problem and proposed distributed optimal solution with low complexity. This solution to the sub-problem is actually an optimal one to the original version for some specific routing trees.

In a more general case, Zheng and Shroff [61] investigated the utility maximization problem for efficient data gathering in large-scale networks constrained by the imposed deadline. The authors considered the general class of utility functions which are monotone submodular. They examined the general optimization problem for both the raw data gathering and in-network aggregation scenarios, and established provable bounds for approximation solutions.

c: EXPLOITING THE SPATIAL DATA CORRELATION

In sensor network, nodes are often densely deployed in the target region to achieve satisfactory coverage. Thus, readings captured by nodes distributed in proximity usually exhibit similarity to some extent, which is known as the spatial correlation. The above works on DCAS [54]–[57] did not account for the redundancy of data sent by nodes in proximity and thus may waste lots of time and energy in gathering data that are not very representative. Alina *et al.* [58] considered utilizing the spatial correlation to further enhance QoA at the sink. They formulated a bi-objective optimization problem which maximized the number of source nodes performing data aggregation as well as the spatial dispersion among the participating nodes. This is an NP-hard problem and they proposed a heuristic distributed solution named SDMAX, which scalarized QoA by assigning weights to the two optimization metrics. The structure of SDMAX bears some similarity to the optimal dynamic programming proposed in [57], but Alina *et al.* did not investigate any performance guarantee for SDMAX.

d: CROSS-LAYER OPTIMIZATION OF ROUTING AND SCHEDULING

The aforementioned works proposed optimal scheduling algorithms under a given aggregation tree. As a matter of fact, the structure of the underlying aggregation tree also plays a vital role in the QoA optimization. Alina *et al.* [59] proved that the ratio between the maximum achievable QoAs acquired under the best routing tree and the worst-case tree can be as large as $O(2^D)$, where D is the deadline imposed by users. They formulated a combinatorial optimization problem which addressed the importance of both the scheduling policy and the tree structure. They proposed a near-optimal algorithm with bounded approximation gap to construct a routing tree. The algorithm is based on an existing framework called Markov approximation [63] and it enables sensor nodes to iteratively migrate towards a near optimum tree in a distributed manner. For scheduling policy, they adopted the very basic framework of the dynamic programming algorithm proposed in [54], but eliminated some concurrent transmissions to avoid interferences aroused by the one-hop interference model in [54].

e: SCHEDULING UNDER PHYSICAL INTERFERENCE MODELS

The approaches above [54], [58], [59] tackled DCAS problem by utilizing the protocol interference model. Yousefi *et al.* [60] took a step forward to consider real-time data aggregation under the SINR model, which culprits the interference in a more accurate way. The problem was proved to be a NP-complete problem and the authors devised a scheduling method based on the Markov approximation framework. They also incorporated the successive interference cancellation (SIC) technique into their solution to further improve QoA. Finally, they obtained the theoretical upper bounds on QoA under the SIC and SINR models.

3) AGGREGATION SCHEDULING WITH THROUGHPUT OPTIMIZATION

Much research attention [46], [64]–[67] has been given to the throughput optimization for multi-channel WSNs. In these works, different frequencies are assigned to links to eliminate interferences in proximity and reduce the duration of data aggregation.

a: SCHEDULING ON ORTHOGONAL CHANNELS

Ghosh *et al.* [64] considered the joint optimization of frequency-time-slot assignment and tree construction. They designed efficient scheduling method which is a constant factor approximation on the optimal network throughput. They considered minimizing the maximum end-to-end delay constrained by the network throughput imposed. The authors also introduced a (10, 7)-bicriteria approximation algorithm to form a spanning tree in which the maximum node degree is bounded by $\Delta^* + 10$, and the network radius is at most 7 times of the minimum possible radius under Δ^*. In [65], Incel *et al.* incorporated the transmit power control into a multi-channel scheduling framework to further mitigate the interference. They revealed that although the power control indeed cuts down the latency for a single frequency channel, applying multiple frequencies can eliminate most of interferences and thus is more effective in enhancing the time efficiency.

Ji *et al.* considered optimizing the capacity for large-scale sensor networks under both the deterministic network model [46] and the probabilistic network model [66]. In [46], they addressed the distributed data gathering issue under the general SINR interference model for the deterministic and asynchronous wireless networks. They examined sensor data collection of two scenarios. For raw data collection, they invented a Distributed Data Collection (DDC) algorithm, which is proved to be scalable and order-optimal in terms of maximum achievable capacity. For aggregate data collection, they devised a Distributed Data Aggregation (DDA) algorithm with bounded delay. The same authors studied the achievable network capacity in a more realistic network containing lossy links [66], where they proposed efficient algorithms with the worst case performance guarantee on the network capacity for both the one-time (snapshot) and continuous data collection.

b: SCHEDULING ON PARTIALLY OVERLAPPLING CHANNELS

The above works [64], [65] employed the Orthogonal Channel (OC) in the multi-channel assignment. Although OC can effectively alleviate the interference, it is a waste of spectrum since among all the 11 available channels in 2.4 GHz ISM band defined by IEEE 802.11b/g standards, only 3 channels are orthogonal. To increase the network throughput, other researchers exploited the Partially Overlapping Channel (POC) which increases the network throughput by tolerating some level of interferences. In these works, two transmissions are considered orthogonal if they are physically separated far away, even if they are spread over the adjacent and overlapping channels. However, POC could not optimally make use of the entire spectrum capacity. Ghods [67] investigated a combination of OCs and POCs to increase the potential parallel transmissions and maximize the data collection rates for continuous surveillance applications. They devised an algorithm which simultaneous executes processes of building the aggregation tree, assigning channels, and performing scheduling. The algorithm is conducted in a top-bottom and level by level manner starting from the sink. In each level, a node with less choices of parents is associated with higher priority to decide its parent, channel, and transmission slots at the same time.

4) AGGREGATION SCHEDULING WITH ENERGY OPTIMIZATION

There are also some works [68]–[74] focusing on designing energy efficient schedules, but the schedules constructed are not guaranteed to have any latency or capacity bound.

a: SCHEDULING BASED ON A SINGLE ROUTING TREE

Hohlt *et al.* [68] devised the flexible power scheduling (*FPS*) algorithm. In *FPS*, each parent node assigns time slots to its child nodes to eliminate collisions from siblings. Simulation results exhibit that *FPS* is energy efficient and can adapt to the workload changes in the network. However, *FPS* cannot prevent collisions coming from nodes with different parents and the schedule constructed is not conflict-free. Wu *et al.* [69] proposed a distributed cross-layer scheduling (*DCS*) mechanism. In *DCS*, every node negotiates its transmission schedule with its parent node. The node then follows the schedule constructed to communicate and go to sleep. The schedule constructed can reduce the idle listening, overhearing, as well as the state transition between sleeping and active statuses. However, *DCS* lacks support for incremental schedule updates in response to changes in the network topology.

Other researchers focused on enhancing the energy efficiency of data gathering, by joint optimizing the routing and scheduling. The aforementioned works utilize a single tree structure for data gathering throughout the network lifetime, which by definition is the period from the very beginning till the first node runs out of battery. Due to the energy cost of receiving packets, the nodes with larger node degrees in a routing tree would consume energy faster than those with smaller node degrees, resulting in unbalanced energy consumption. One potential approach to deal with this is to build multiple routing trees with corresponding schedules and try to use different trees for routing at different sampling intervals.

b: SCHEDULING BASED ON MULTIPLE ROUTING TREES

Kalpakis *et al.* [70] proposed an integer program to find the optimal network flow for solving the maximum lifetime data aggregation problem. They designed a heuristic algorithm to obtain a good approximation of the optimal flow in polynomial time. The derived flow is further decomposed into a series of spanning trees and every spanning tree is allocated a time span denoting the period that the tree is used for routing. This heuristic algorithm performs well with respect to the network lifetime, but is computationally expensive especially for the large scale sensor networks. In a later work [71], the authors proposed a cluster-based solution to enhance the scalability of the algorithm.

Lee and Keshavarzian [72] built a set of routing trees to optimize the network lifetime. They assumed all nodes are equipped with equal initial energy. Their proposed approach contains three phases. The first phase concerns assigning a layer to each node to form a hierarchical structure in the network. All links among nodes at the same layer are then removed from the connectivity graph. The second phase solves the maximum lifetime data gathering problem using linear programming. The third phase constructs a certain number of routing trees to approximate the optimal flow derived. The routing trees constructed are selected to be used as the routing structures at different sampling intervals in a round-robin way.

The above two works [70], [72] have exploited both the temporal and spatial load balancing by forming multiple routing trees and switching to different routing trees over sampling intervals. Although these methods can better balance the traffic load among sensor nodes than using a single routing tree, they require a data collection schedule to be constructed and recorded for each routing tree formed, thus increase the communication cost and the storage cost.

c: SCHEDULING BASED ON RINGS OVERLAY

For the scheduling works based on the tree structures, a single link failure would result in the data loss of the entire subtree. To enhance the robustness of data gathering, other researchers [73], [74] based their scheduling works on the multi-path routing structures, where every sensor node can have several parents to forward copies of a single piece of data. Data can be successfully delivered to the sink provided that one of the propagation routes is failure-free. Hai and Tang [73] utilized the rings overlay, a special case of multi-path routing structure, to make better use of the broadcast feature of wireless communication and reduce the communication failures. They proposed a distributed approach to construct the rings overlay using only local neighborhood information of sensor nodes. Later, the same authors put forward a distributed scheduling method to build a single schedule based on the rings overlay for communication [74]. The scheduling method incurs very low overheads in terms of the run time and message complexity during the execution of scheduling algorithm. This is achieved by fixing the relative scheduling order of nodes before the scheduling starts, so that nodes do not need to compete for the channel access. Then, the authors derived a theoretical lower bound on the shortest possible latency.

5) AGGREGATION SCHEDULING WITH SECURITY ENHANCEMENT

Besides optimizing the delay and energy efficiency, Kirton *et al.* [75] exploited aggregation scheduling for the security purpose. They designed TDMA scheduling algorithms to protect the Source Location Privacy (SLP) in a sensor network deployed for the asset monitoring, where data aggregation is triggered when a node called source gets aware of the existence of a particular asset. To provide SLP, it is essential to prevent attackers from tracing back the source and capturing the asset. Most methods achieved this by altering the routing layer to generate a path that diverts an attacker away from the source. Kirton *et al.* provided SLP in the MAC layer by achieving a similar traffic alteration with much less message overhead. They put forward novel formalisation of different classes of attackers, and the SLP-aware data aggregation schedules. Then, a decision procedure similar to model checking is presented to check whether a given schedule is SLP-aware. Finally, a 3-stage distributed algorithm is proposed to transform an original schedule into a SLP-aware schedule against a particular sort of eavesdroppers.

C. TDMA SCHEDULES FOR NON-AGGREGATE DATA GATHERING

Both the link scheduling and aggregate scheduling algorithms discussed above could not satisfy the communication requirements of non-aggregate data gathering, where different nodes forward different numbers of packets upstream (i.e., sensor nodes near to the sink need to transmit more data packets than nodes farther away from the sink). To tackle this problem, many non-aggregate scheduling algorithms are designed. Most of these algorithms are devised to handle the full traffic pattern, where every sensor node produces one data packet to forward to the sink.

Features of Non-aggregate Data Gathering:

- The number of time slots allocated to an intermediate node is proportional to the number of its descendants.
- The first transmission slot of an intermediate node does not need to be arranged after all the transmission slots of its child nodes.

Next, we categorize works [76]–[92] on the non-aggregate scheduling mainly based on the design objective.

1) THROUGHPUT OPTIMIZATION

Some works on the non-aggregate scheduling aim to optimize the throughput at the base station. Ahn *et al.* [76] devised the *funneling-MAC* protocol to achieve this goal. Funneling-MAC carries out TDMA scheduling in a region near to the base station (called the intensity region) to ensure quick and reliable data transmissions, while implementing the CSMA protocol in the rest of the network to offer the flexibility. *Funneling-MAC* naively assumes that the base station can apply power control to communicate directly with the nodes in the intensity region. This assumption is not always true due to the weak transmission power of the base station or some obstacles that block the routes from the sink to sensor nodes in the intensity region. Song *et al.* [77] put forward a distributed TDMA protocol named *TreeMAC*, which constructs a TDMA schedule to enable each node to get an opportunity to access to the channel proportional to its traffic demand. *TreeMAC* cuts time into discrete frames, each of which is composed of three time slots. A parent node assigns frames to its child nodes, and every child node determines its slot number on the basis of its level in the data collection tree. The frame assignment eliminates the collisions from sibling nodes in the horizontal direction, while the slot assignment eliminates the collisions from two hop neighbors in the vertical direction. The authors proved that the *TreeMAC* can offer a throughput guarantee of at least 1/3 of the optimum. Simulation results show that *TreeMAC* outperforms *funneling-MAC* in achieving higher network throughput.

2) MINIMIZING THE SCHEDULE LENGTH

Other non-aggregate scheduling algorithms target at minimizing the length of the schedule.

a: BASIC STRATEGIES

Song *et al.* [78] proposed *STREE*, a distributed and time-optimum packet scheduling algorithm. *STREE* reduces the latency by letting all the one-hop subtrees perform data forwarding simultaneously. One-hop subtrees refer to the subtrees rooted at the child nodes of the base station. In *STREE*, the sink collects data packets from its child nodes in turns in a decreasing order of the subtree size. In each one-hop subtree, there are many data propagation paths from leaf nodes to the root, but at most one data propagation path is actively transmitting data at a time. For a path that is active, time slots are allocated to nodes so that the even-hop nodes and the odd-hop nodes transmit alternatively. *STREE* can provide latency bounds regardless of whether every sensor node produces the same volume of data or different nodes generate heterogeneous amount of data.

Choi *et al.* [79] formulated the non-aggregate data scheduling problem as the *Minimum Information Gathering Time Problem* (MIGTP). They built a routing tree and constructed a minimum-latency data collection schedule based on the tree. The authors proved that *MIGTP* is NP-complete on general graphs, by reducing it from the classical *partition problem* [62]. They proposed heuristic algorithms for the line and tree topologies respectively. The authors proved that the heuristic algorithms offer latency bound of $3N - 3$ time slots, which is optimum. The authors also proposed a heuristic algorithm for a general network, by building a minimum spanning tree and then trimming the tree edges so that transmissions in various one-hop subtrees do not conflict with each other and can be scheduled in parallel.

One of the early works on the non-aggregate scheduling is done by Florens and McEliece [80], Florens *et al.* [81], Florens and McEliece [82]. They addressed the problem of scheduling the packet distribution from the base station to sensor nodes, and argued that it can be regarded as an inverse problem of data collection. Centralized algorithms were proposed to compute minimum-latency schedules for some special network topologies. For the line network, the key strategy is to make the base station first deliver data packets destined to the farthest node in the network, then transmit packets to the second farthest node, and so forth. A node between the base station and the destination of the packet relays the packet in the next time slot on arrival of the packet. The upper part of Fig. 6 represents an optimal schedule that uses 11 time slots for distributing 5 packets in a 10-node line network. The figure shows that the data transmissions are performed as quickly as possible without causing collisions. Once the optimal schedule for the packet distribution is found, the schedule for data collection can be derived accordingly, as shown in the lower part of Fig. 6. This basic idea of link networks can also be applied to multi-line and tree networks.

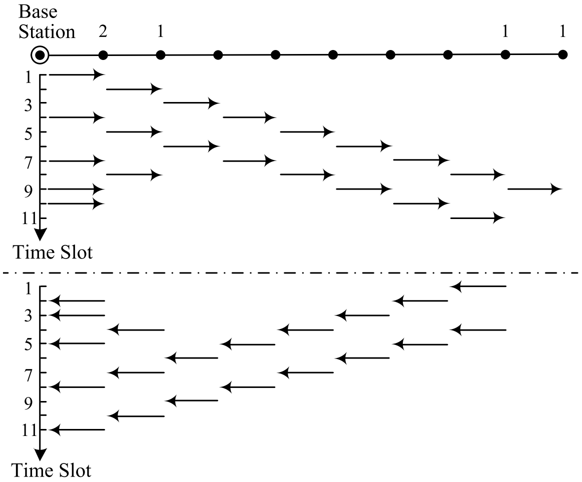
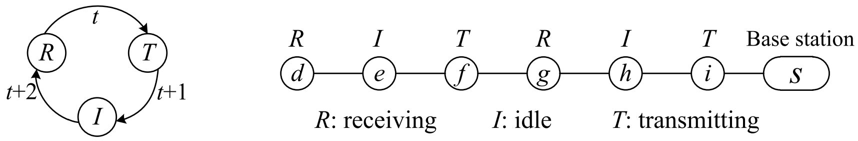


FIGURE 6. The optimal time scheduling for a 10-node line network with the minimum schedule length 11 slots (taken from [80]).



(a) State transition of the nodes. (b) A linear network with initial state assignment.

FIGURE 7. The state transition of nodes and the initial state assignment for a linear network (taken from [83]).

Florens' scheduling algorithms [80]–[82] are centralized because the schedules are computed at the base station. Gandham *et al.* [83] proposed the distributed version of the Florens' algorithms. They aimed to build the minimum-latency schedule to collect raw data in a network in which each node produces exactly one data packet in each sampling instance. The authors proved that the proposed distributed scheduling algorithm uses at most $3N$ time slots for an N node network. Besides minimizing the schedule length, the proposed method also accounts for reducing the storage burden on the nodes by restricting each node to buffer at most two packets. Since Gandham's algorithm will be compared against the scheduling method we shall propose, we illustrate it in more details below.

For line networks, every sensor node is allocated an original status according to its hop distance away from the base station. As shown in Fig. 7 (b), a node that is h hops from the sink node is assigned status T (transmitting), if h mod 3 is 1; status I (idle), if h mod 3 is 2 and status R (receiving), if h mod 3 is 0. The node statuses repeatedly changes among the three states shown in Fig. 7 (a). As a result, for each node in status R, there is only one neighbor in status T, so that the packet transmission is collision free and one packet arrives at the base station each 3 time slots. The basic thoughts can be extend to more complex topologies, for instance the multi-line networks, tree networks and general networks. For multi-line networks, transmissions are first scheduled in the line with the largest number of remaining packets. For tree networks, transmissions are scheduled concurrently along multiple one-hop subtrees. For general networks, a Breadth First Search (BFS) tree is first constructed after that the scheduling method is implemented based upon the tree structure. Most importantly, in order to make the distributed algorithm work, necessary information must be exchanged in the initialization phase. Every sensor node must be aware of the ID of its one-hop subtree, the number of nodes in all the other one-hop subtrees, and the conflict map of the network, but all nodes in the network including the sink do not need to realize the complete network topology. In fact, the initialization phase takes $3N + k$ number of time slots, where N is the network size and k is the number of the one-hop subtrees.

b: COPING WITH SPECIFIC TRAFFIC PATTERNS

In the follow-up work of the same authors, they extended their algorithm to construct a minimum-latency schedule tailored to a given traffic pattern [84], in which not every node gets data to forward. For those nodes which indeed have data, they may generate lots of data that need to be transmitted in multiple packets. The resultant algorithm called *SPARSE* requires additional knowledge of the traffic pattern in order to construct the schedule. To recognize the traffic pattern in the network, information needs to stream over the whole routing tree to accomplish an in-order tree traversal phase. Then, the base station must disseminate the relevant information of traffic pattern to each sensor node in the network. Finally, each node runs the distributed scheduling algorithm to build a best-fit schedule for the traffic pattern given, and the data gathering proceeds with the schedule constructed.

SPARSE enables every sensor node to buffer utmost two data packets. Other scheduling algorithms work without this limit. Ergen and Varaiya [85] tried to minimize the schedule length and proved that the problem is NP-complete by reducing it from the Graph Coloring problem. The authors argued that the difficulty of the problem lies in that multiple subsets of non-interfering nodes are possible to be scheduled in each time slot. The chosen of a subset in one slot directly influences the set of nodes available to be scheduled in the next time slot. A node-based and a level-based scheduling heuristic algorithms were proposed based on graph coloring. It was shown that the node-based scheduling algorithm works better when the routing tree has uniform packet density, or the low layers of the routing tree have higher density of packets, whereas the level-based scheduling algorithm performs better when the upper levels of the tree have higher packet density.

Ergen and Varaiya [86] proposed the *PEDAMACS*, a power efficient and delay aware TDMA protocol. It assumes all the packets generated in the network are destined to the same node named the access point (AP). The AP has abundant transmission power to get access to all the other nodes in one-hop. Initially, each node discovers its neighbors and its interferers, and its parent in the tree for data gathering. Then, each node reports this information to the AP. On collecting all the information, AP constructs and announces the transmission schedule to all the nodes in the network. The schedule construction follows the method discussed in [87]. In addition, upon changes of the network topology, *PEDAMACS* enables nodes to piggyback the new topology information in data packets transmitted to the base station, thereby reducing the overhead in explicitly relearning the whole network topology.

c: APPLYING BATCH PROCESSING

All the above works assume that a data packet can accommodate one sensor reading only. In practice, the sensor reading often has small size and multiple sensor readings could be fit into one data packet for transmission. Paradis and Han [88] proposed the *TIGRA*, a distributed scheduling algorithm that exploits batch processing to reduce the overhead in transmitting packet headings. The batch processing allows at most m readings to be concatenated or combined at internal nodes and be delivered upstream as one packet over the data collection tree. To make best use of batch processing, the number of 'saturated' packets that have m sensor readings should be maximized. *TIGRA* devises a method to accumulate the sensor readings received at an intermediate node to fill up one data packet. *TIGRA* utilizes a graph coloring mechanism to build a collision-free schedule that offers near-optimal latency for data collection.

d: SCHEDULING IN DUTY-CYCLED NETWORKS

All the aforementioned scheduling algorithms [76]–[86] are developed for non-duty-cycled WSNs, in which sensors are always well prepared to transmit or receive data packets. In these networks, the process of the packet distribution from the sink to sensor nodes can be regarded as an inverse process of data gathering. Thus, the symmetry property is valid in a non-duty-cycled WSN. By contrast, this symmetry is not valid in a duty-cycled WSN, in which sensor nodes periodically switch to the sleeping mode to save batteries, and a data packet can be successfully forwarded to the destination only when both the sender and receiver are active. Shen *et al.* [89] extended the work of Florens and McEliece [80], Florens *et al.* [81], Florens and McEliece [82] by considering the fast data gathering in a linear and duty-cycled WSN. They proposed an optimal algorithm and a distributed method under the assumption that every node only works at one time slot in a cycle in the duty-cycled mode. Both algorithms proposed are proved to have bounded performance gap to the optimal performance derived in the non-duty-cycled scenario.

e: APPLYING MULTIPLE ROUTING STRUCTURES

The scheduling works discussed above utilized a single routing structure for data collection. There are other works formed a set of routing infrastructures to prolong the network lifetime. In these works, one schedule needs to be constructed for each routing infrastructure. Lee *et al.* [90] investigated the establishment of the lifetime-optimal DAG. In their work, maximizing the lifetime is first formulated as a linear programming problem. Then, the authors obtained the optimal DAG structure on the basis of the linear formulation. The derived DAG structure depicts the orientations and the amount of the traffic flow which equalize the overall energy expenditure across different sensor nodes. After that, a collection of sub-DAGs are decomposed to approximate the optimal DAG. Every sub-DAG logged the quantity of packets a node sends to each parent node in one sampling interval such that the average use ratio of each edge for sending data is approximately identical to the optimal use ratio imposed by the lifetime-optimal DAG. Sensor nodes transition to different sub-DAGs for routing over various sampling intervals. By taking advantages of both the spatial and temporal load balancing, the proposed method vastly prolongs the network lifetime. Nevertheless, they require a data collection schedule to be constructed and recorded for each sub-DAG formed and thus incur large storage and computation overhead.

f: SCHEDULING FOR UNDERWATER ACOUSTIC NETWORKS

In addition to the theoretical works, other researches focused on using TDMA techniques to facilitate applications such as the under water communications. Zhang *et al.* [91] have studied the throughput maximization problem in a general underwater communication scenario. They have taken into consideration of both a practical network topology and the mobility of sensor nodes. In their work, the topology of communication network is formulated as a three dimensional (3D) scenario, which indicates a more complicated interference scenario. In their work, each transmission frame contains K time slots, and a central node performs scheduling and broadcasts the schedule to sensor nodes once per frame. Initially, the central node gathers the information of every communication edge via a control channel. The information contains the velocities and locations of the message senders and receivers, as well as the amount of traffic to be flowed over each link. For each time slot, the central node first constructs an interference graph according to the estimated positions of nodes that are currently communicating, as well as a preset threshold. After that, the central node performs scheduling that selects the maximum subset of nodes to transmit simultaneously in a time slot so that the network throughput gets optimized. The resultant transmission table is then broadcasted from the central node to the rest of network per frame, following which all nodes involved start sending and receiving packets.

Liao *et al.* [92] have devised a MAC protocol named DTSM (Distributed Traffic-based Scheduling MAC) to enhance the network throughput for underwater the acoustic sensor networks. DTSM works by performing joint bandwidth optimizing and media access control. The key idea is to allocate bandwidth to sensor nodes based on their traffic loads, so that the nodes with larger amount of traffic can be assigned more transmission bandwidth. Specifically, the scheduling process is conducted on the basis of ages of data packets, indicating that the older packets gets scheduled earlier. DTSM has incorporated a RTS/CTS handshake framework for every sensor node to determine the age of packets of other nodes in a distributed manner. This is achieved by dividing each time slot into multiple mini-slots for RTS/CTS exchange. DTSM is proved to achieve reasonable bandwidth allocation and high channel utilization.

IV. COPING WITH TRAFFIC DYNAMICS IN SENSOR DATA COLLECTION

All the scheduling algorithms (including link scheduling, aggregate scheduling and non-aggregate scheduling) discussed above are designed to handle the static traffic patterns only. However, in many applications, the traffic pattern naturally exhibits the dynamic and non-deterministic characteristic, either due to the nature of the applications or the energy conservation concerns. Static schedules are not suitable to deal with the dynamic traffic patterns in the network, because a schedule constructed for a heavy traffic pattern may cause time and energy waste in handling a light traffic pattern, while a schedule built for a light traffic pattern cannot satisfy the communication demands of a heavy traffic pattern. One possible approach is to construct and deploy a new TDMA schedule tailored to the new traffic pattern whenever the traffic pattern changes. However, identifying new traffic patterns and disseminating new schedules over the network both require sensor nodes to communicate with each other, which introduces extra energy and latency overhead that are very likely to cancel out or even outweigh the benifits of deploying new TDMA schedules, particularly when the traffic pattern changes frequently.

Some existing protocols [93]–[99] have emphasized the scheduling problem for dynamic traffic patterns in the network. A common method is to periodically exchange the information of traffic load among sensor nodes and adjust the TDMA schedule accordingly. The Traffic-Adaptive Medium Access Protocol (*TRAMA*) devised by Rajendran *et al.* [93] falls into this category. *TRAMA* divides time into two phases: the random-access phase and the schedule-access phase. In the random-access phase, every sensor node broadcasts messages of nodes in its neighborhood, and learns its two hop neighbors by receiving packets containing the neighborhood information from its one-hop neighbors. In the schedule-access period, the potential message senders first broadcast their schedule information, by providing their neighbor nodes with an latest list of receivers for packets currently in the transmission queues of the senders. Based on this information, the nodes execute a distributed scheduling algorithm to decide the senders and receivers for each time slot in the schedule-access period. Sensor nodes can go to sleep when they are not actively receiving or dispatching data. The strength of *TRAMA* is that it is energy efficient and can facilitate slot reuse. However, *TRAMA* incurs non-negligible communication overhead in periodically exchanging traffic statistics to reflect the dynamic traffic patterns in the network. In addition, *TRAMA* sacrifices the latency for energy efficiency. In *TRAMA*, packets have to be buffered and cannot be transmitted until the schedule is announced. This introduces extra latency in data collection.

Another way to make the schedule adaptive to the dynamic traffic pattern is through the use of hybrid protocols, which combine the strengths of the TDMA and CSMA protocols. Rhee *et al.* [94] proposed the Z-MAC protocol that can reduce the latency in data collection under dynamic workload. Z-MAC uses a distributed randomized algorithm called *DRAND* [95] to build a TDMA schedule. A node arranged to transmit in a time slot is named an owner of that slot and the other nodes are called the stealers. Before a node transmits during a slot, it probes the channel and delivers a packet only if the channel is unblocked. However, the owners of a time slot always have higher priorities in accessing the channel. If the owners do not transmit, the stealers can steal the slot for transmission. Thus, in Z-MAC, a node may send data in any time slot and the latency is reduced. However, this protocol is not energy efficient since each node must turn on its radio to listen to the channel at any time for possible transmissions.

The *ASAP* protocol proposed by Gobriel *et al.* [96] further improves Z-MAC in terms of energy efficiency. *ASAP* is based on the tree topology. The *ASAP* protocol requires the stealers of a time slot to be the child nodes of the owners, which means only sibling nodes contend the channel for early transmissions. This enables a parent node to power off its radio to go to sleep on hearing a vacant slot, which indicates neither the owner nor the stealers have data to transmit. *ASAP* outperforms Z-MAC in the energy efficiency but the latency is still high, since the transmission ahead of schedule only happens among nodes at the same level. The latency of transmissions across the network is not improved. In addition, the energy saving of *ASAP* is rather limited, since each node has to wake up quite often to receive the potential data packets and check whether it can get chance to send earlier. Extra energy spent on idle listening and overhearing compromise the energy efficiency to some extent.

Chipara *et al.* [97] proposed the dynamic conflict-free query scheduling (*DCQS*) scheme to handle the dynamic traffic pattern caused by the injection of new queries and the deletion of old queries in the network. *DCQS* works by constructing a latency-optimized schedule for data collection in each query. It also computes a minimum query inter-release time for successive query instances to ensure that the data delivering executed in a slot are collision-free. Thus, *DCQS* adapts to the dynamic workloads without explicitly reconstituting the schedule. Since it only requires the local information to build the transmission schedule and obtain the query inter-release time, *DCQS* can cope with the topology changes. However, *DCQS* can only handle the dynamic workload caused by query injections and deletions. It does not consider the dynamic workload in executing each query.

The Z-MAC [94], *ASAP* [96], and *DCQS* [97] schedules that adapt to the dynamic traffic patterns are designed to collect aggregate data only. Other works [98], [99] addressed a unique problem to design an energy and delay efficient TDMA schedule that gracefully handles the dynamic and non-aggregate traffic in the network.

Zhao and Tang [98] proposed a Traffic Pattern Oblivious (TPO) scheduling method to boost the time efficiency and energy efficiency of continuous sensor data gathering, which periodically gathers the raw data captured by individual sensor nodes. Traffic patterns in continuous data gathering usually exhibit the dynamically changing features owing to the energy conservation concerns. TPO facilitates each node to transmit all data in its consecutive sending slots regardless of the network traffic pattern. Thus, once a receiver discovers one transmission slot of a sender is left empty, the receiver is convinced that no more data would arriving from that sending node in the current sampling interval, and it can safely stop listening to the sender without missing any data. The energy expended by every sensor node self-adjusts to its required amount of work imposed by any traffic pattern. Another benefit is the reduced latency of data gathering, since the sink can conclude data gathering once it has performed one time of idle listening to each of its child node.

TPO deals with the tree routing structure only, and it did not take a step forward to optimize the routing infrastructures to boost the performance of data gathering. In their later work, Zhao *et al.* [99] extended their scheduling works on the DAG (Directed Acyclic Graph) routing structure, where every internal node can have a number of parent nodes and child nodes. Their aim is to extend the network lifetime for as long as possible under dynamic traffic patterns, and the network lifetime is regarded as the period from the very beginning till the first node uses up its battery. The authors tried to build a unique routing structure to evenly distribute traffic loads among all nodes in the network. In their work, searching for the lifetime optimized DAG is formulated as a mixed integer programming problem. Then, they presented a heuristic method to derive a near-lifetime-optimal DAG, based on which a single transmission schedule is constructed and is consistently utilized for data gathering throughout the life cycle of the network. At every sampling interval, rather than letting every node deliver data to a stationary parent, the schedule allows every sensor node to dynamically pick diverse parents as targets for delivering various packets. The choice of parents can always guarantees the actual amount of traffic flow on each edge approximate the flows described in the DAG structure.

V. FUTURE WORKS

The work conducted in this review has several limitations. It does not generalize those efforts made in emerging networks, such as the VANET (Vehicular Ad-hoc NETwork) systems. Due to space limitation, this paper only incorporates works conducted in sensor data gathering. There are other works that conduct packet schedulings in a TDMA channel to enhance the freshness of information. These works are related to the field of control theory and are excluded in the discussion of this paper. Besides the existing scheduling mechanisms summarized, other possible works include:

A. COPE WITH TOPOLOGY CHANGES CAUSED BY NODE FAILURES OR NEW NODES JOINING THE NETWORK

Most scheduling algorithms work in wireless sensor networks in which the network topologies are static. In practice, the network topology may not always be stable and could change over time due to node failures or new nodes joining the network. One possible approach to cope with it is to force all sensor nodes to run the scheduling algorithm from scratch again to build a new schedule. Due to the communication overhead involved, this method may be effective only when the network experiences a severe topology change, i.e., a large portion of nodes join or leave the network. On the other hand, incrementally adjusting the existing schedule could be a better choice for dealing with minor topology changes. Localized algorithms can be designed to let the schedule adjustment be made only by the nodes in the local area that experiences the topology change.

B. CONSIDER LINK FAILURES

The connectivity graph formed in most existing works includes only those highly reliable links. To deal with the temporally changing link quality, it would be useful to model the link condition in the problem formulation in the first place. The aim is to enable each node selects its parent opportunistically, to reduce latency and ensure a good link quality. Also, to increase the transmission reliability, one time slot can be extended to be long enough to accommodate several rounds of transmissions and retries required by a three-way handshake agreement. Or, duplicate transmissions can be performed through multiple disjoint journeys. An original packet is considered correctly received as long as one of those copies reaches the destination. It is of interesting to conduct a study of the tradeoff between communication cost and the reliability performance.

C. DESIGN DATA APPROXIMATION TECHNIQUES IN THE APPLICATION LAYER

Approximate data collection is an energy conservation strategy that trades the data accuracy for energy efficiency. It will be interesting to study and design data approximation techniques that can meet the user-designated data precision constraints while reducing the energy consumption and latency of data collection as much as possible. In addition, it would be useful to exploit data approximation techniques for improved quality of monitoring results.

VI. CONCLUSION

In wireless sensor networks, Median Access Control (MAC) protocols are designed to appropriately handle the concurrent transmissions from multiple nodes and reduce the effect of collisions. This paper reviews two basic categories of MAC protocols: contention-based protocols and schedule-based protocols. Our focus is the TDMA scheduling protocols, which is schedule-based and widely used in wireless sensor networks. Compared with contention-based methods, the amounts of energy consumption and latency involved in data gathering with TDMA protocols are bounded.

It should be emphasized that compared with other surveys in this domain, this review does not confine itself to deal with collecting a particular form of data, but provides a unified framework to integrate the data semantics in a broader sense into the design of TDMA scheduling algorithms. In doing so, we classify TDMA scheduling into three categories according to different communication patterns in the network. That is, the link scheduling, aggregate scheduling and non-aggregate scheduling. We have elaborated works in each category and provided a detailed briefing on how the TDMA schedules handle the network traffic dynamics.

ACKNOWLEDGEMENT

The authors thank the Staff in the School of Aerospace Science and Technology, Xidian University. They also thank the anonymous reviewers for their constructive comments.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.

[2] *Exposed Terminal Problem*. [Online]. Available: https://en.wikipedia.org/wiki/Exposed_node_problem

[3] A. Sgora, D. J. Vergados, and D. D. Vergados, "A survey of TDMA scheduling schemes in wireless multihop networks," *ACM Comput. Surv. (CSUR)*, vol. 47, no. 3, p. 53, 2015.

[4] N. Abramson, "The ALOHA system: Another alternative for computer communications," in *Proc. AFIPS*, 1970, pp. 281–285.

[5] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Standard 802.11b-1999, 1999.

[6] *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, Standard 802.15.4, 2003.

[7] J. Jubin and J. D. Tornow, "The DARPA packet radio network protocols," *Proc. IEEE*, vol. 75, no. 1, pp. 21–32, Jan. 1987.

[8] P. Karn, "MACA—A new channel access method for packet radio," in *Proc. 9th ARRL Comput. Netw. Conf.*, 1990, pp. 134–140.

[9] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. IEEE INFOCOM*, Jun. 2002, pp. 1567–1576.

[10] T. Van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proc. ACM SenSys*, 2003, pp. 171–180.

[11] A. El-Hoiydi and J.-D. Decotignie, "WiseMAC: An ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks," in *Proc. 9th Int. Symp. Comput. Commun. ISCC*, Jun. 2004, pp. 244–251.

[12] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. 2nd Int. Conf. Embedded Netw. Sensor Syst. SenSys*, 2004, pp. 95–107.

[13] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proc. 4th Int. Conf. Embedded Networked Sensor Syst. SenSys*, 2006, pp. 307–320.

[14] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493–506, Jun. 2004.

[15] J. L. Hill and D. E. Culler, "Mica: A wireless platform for deeply embedded networks," *IEEE Micro*, vol. 22, no. 6, pp. 12–24, Nov. 2002.

[16] R. Laufer and L. Kleinrock, "The capacity of wireless CSMA/CA networks," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1518–1532, Jun. 2016.

[17] S. Wang, V. Venkateswaran, and X. Zhang, "Fundamental analysis of full-duplex gains in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1401–1416, Jun. 2017.

[18] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh, "Deploying a wireless sensor network on an active volcano," *IEEE Internet Comput.*, vol. 10, no. 2, pp. 18–25, Mar. 2006.

[19] N. Javaid, I. Israr, M. A. Khan, A. Javaid, S. H. Bouk, and Z. A. Khan, "Analyzing medium access techniques in wireless body area networks," *Res. J. Appl. Sci., Eng. Technol.*, vol. 7, no. 3, pp. 603–613, Jan. 2014.

[20] S. Ramanathan, "A unified framework and algorithm for channel assignment in wireless networks," *Wireless Netw.*, vol. 5, no. 2, pp. 81–94, Mar. 1999.

[21] S. Gandham, M. Dawande, and R. Prakash, "Link scheduling in sensor networks: Distributed edge coloring revisited," in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Societies*, Mar. 2005, pp. 2492–2501.

[22] W. Wang, X.-Y. Li, O. Frieder, Y. Wang, and W.-Z. Song, "Efficient interference-aware TDMA link scheduling for static wireless networks," in *Proc. 12th Annu. Int. Conf. Mobile Comput. Netw. MobiCom*, 2006, pp. 262–273.

[23] D. M. Blough, G. Resta, and P. Santi, "Approximation algorithms for wireless link scheduling with SINR-based interference," *IEEE/ACM Trans. Netw.*, vol. 18, no. 6, pp. 1701–1712, Dec. 2010.

[24] Y. Zhou, X.-Y. Li, M. Liu, X. Mao, S. Tang, and Z. Li, "Throughput optimizing localized link scheduling for multihop wireless networks under physical interference model," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 10, pp. 2708–2720, Oct. 2014.

[25] J. Yu, B. Huang, X. Cheng, and M. Atiquzzaman, "Shortest link scheduling algorithms in wireless networks under the SINR model," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2643–2657, Mar. 2017.

[26] W. Bai, H. Wang, X. Shen, and R. Zhao, "Link scheduling method for underwater acoustic sensor networks based on correlation matrix," *IEEE Sensors J.*, vol. 16, no. 11, pp. 4015–4022, Jun. 2016.

[27] J. Ma, W. Lou, and X.-Y. Li, "Contiguous link scheduling for data aggregation in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 7, pp. 1691–1701, Jul. 2014.

[28] Y. Wu, X.-Y. Li, Y. Liu, and W. Lou, "Energy-efficient wake-up scheduling for data collection and aggregation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 2, pp. 275–287, Feb. 2010.

[29] C. Berge, *Graphs Hyper Graphs*. Amsterdam, The Netherlands: North-Holland, 1973.

[30] P. Cardieri, "Modeling interference in wireless ad hoc networks," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 4, pp. 551–572, 4th Quart., 2010.

[31] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer, "Complexity in geometric SINR," in *Proc. 8th ACM Int. Symp. Mobile ad hoc Netw. Comput. MobiHoc*, 2007, pp. 100–109.

[32] T. Moscibroda, R. Wattenhofer, and A. Zollinger, "Topology control meets SINR: The scheduling complexity of arbitrary topologies," in *Proc. 7th ACM Int. Symp. Mobile ad hoc Netw. Comput. MobiHoc*, 2006, pp. 310–321.

[33] G. Brar, D. M. Blough, and P. Santi, "Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks," in *Proc. 12th Annu. Int. Conf. Mobile Comput. Netw. MobiCom*, 2006, pp. 2–13.

[34] K. Chen, M. Ma, E. Cheng, F. Yuan, and W. Su, "A survey on MAC protocols for underwater wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1433–1447, 3rd Quart., 2014.

[35] M. Bagaa, Y. Challal, A. Ksentini, A. Derhab, and N. Badache, "Data aggregation scheduling algorithms in wireless sensor networks: Solutions and challenges," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1339–1368, 3rd Quart., 2014.

[36] X. Chen, X. Hu, and J. Zhu, "Minimum data aggregation time problem in wireless sensor networks," in *Mobile Ad-Hoc and Sensor Networks* (Lecture Notes in Computer Science), vol. 3794. Berlin, Germany: Springer, 2005, pp. 133–142.

[37] S. C.-H. Huang, P.-J. Wan, C. T. Vu, Y. Li, and F. Yao, "Nearly constant approximation for data aggregation scheduling in wireless sensor networks," in *Proc. IEEE INFOCOM 26th IEEE Int. Conf. Comput. Commun.*, May 2007, pp. 366–372.

[38] B. Yu, J. Li, and Y. Li, "Distributed data aggregation scheduling in wireless sensor networks," in *Proc. IEEE INFOCOM 28th Conf. Comput. Commun.*, Apr. 2009, pp. 2159–2167.

[39] X. Xu, X.-Y. Li, X. Mao, S. Tang, and S. Wang, "A delay-efficient algorithm for data aggregation in multihop wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 1, pp. 163–175, Jan. 2011.

[40] M. Bagaa, M. Younis, D. Djenouri, A. Derhab, and N. Badache, "Distributed low-latency data aggregation scheduling in wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 11, no. 3, pp. 49:1–49:36, 2015.

[41] Q. Chen, H. Gao, S. Cheng, J. Li, and Z. Cai, "Distributed non-structure based data aggregation for duty-cycle wireless sensor networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2017, pp. 1–9.

[42] H. Yousefi, M. Malekimajd, M. Ashouri, and A. Movaghar, "Fast aggregation scheduling in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 6, pp. 3402–3414, Jun. 2015.

[43] Y. Li, L. Guo, and S. K. Prasad, "An energy-efficient distributed algorithm for minimum-latency aggregation scheduling in wireless sensor networks," in *Proc. IEEE 30th Int. Conf. Distrib. Comput. Syst.*, Jun. 2010, pp. 827–836.

[44] P.-J. Wan, S. C.-H. Huang, L. Wang, Z. Wan, and X. Jia, "Minimum-latency aggregation scheduling in multihop wireless networks," in *Proc. 10th ACM Int. Symp. Mobile ad hoc Netw. Comput. MobiHoc*, 2009, pp. 185–194.

[45] X.-Y. Li, X. Xu, S. Wang, S. Tang, G. Dai, J. Zhao, and Y. Qi, "Efficient data aggregation in multi-hop wireless sensor networks under physical interference model," in *Proc. IEEE 6th Int. Conf. Mobile Adhoc Sensor Syst.*, Oct. 2009, pp. 353–362.

[46] S. Ji and Z. Cai, "Distributed data collection in large-scale asynchronous wireless sensor networks under the generalized physical interference model," *IEEE/ACM Trans. Netw.*, vol. 21, no. 4, pp. 1270–1283, Aug. 2013.

[47] M. K. An, N. X. Lam, D. T. Huynh, and T. N. Nguyen, "Minimum latency data aggregation in the physical interference model," *Comput. Commun.*, vol. 35, no. 18, pp. 2175–2186, Nov. 2012.

[48] H. Li, C. Wu, Q.-S. Hua, and F. C. M. Lau, "Latency-minimizing data aggregation in wireless sensor networks under physical interference model," *Ad Hoc Netw.*, vol. 12, pp. 52–68, Jan. 2014.

[49] N.-T. Nguyen, B.-H. Liu, V.-T. Pham, and T.-Y. Liou, "An efficient minimum-latency collision-free scheduling algorithm for data aggregation in wireless sensor networks," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2214–2225, Sep. 2018.

[50] N.-T. Nguyen, B.-H. Liu, S.-I. Chu, and H.-Z. Weng, "Challenges, designs, and performances of a distributed algorithm for minimum-latency of data-aggregation in multi-channel WSNs," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 1, pp. 192–205, Mar. 2019.

[51] Q. Chen, H. Gao, Z. Cai, L. Cheng, and J. Li, "Energy-collision aware data aggregation scheduling for energy harvesting sensor networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2018, pp. 117–125.

[52] K. Chen, H. Gao, Z. Cai, Q. Chen, and J. Li, "Distributed energy-adaptive aggregation scheduling with coverage guarantee for battery-free wireless sensor networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2019, pp. 1018–1026.

[53] L. N. Balico, A. A. F. Loureiro, E. F. Nakamura, R. S. Barreto, R. W. Pazzi, and H. A. B. F. Oliveira, "Localization prediction in vehicular ad hoc networks," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2784–2803, 4th Quart., 2018.

[54] S. Hariharan and N. B. Shroff, "Maximizing aggregated information in sensor networks under deadline constraints," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2369–2380, Oct. 2011.

[55] S. Hariharan and N. B. Shroff, "Deadline constrained scheduling for data aggregation in unreliable sensor networks," in *Proc. Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw.*, May 2011, pp. 140–147.

[56] S. Hariharan and N. B. Shroff, "On optimal energy efficient convergecasting in unreliable sensor networks with applications to target tracking," in *Proc. 12th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. MobiHoc*, 2011, pp. 1–10.

[57] S. Hariharan, Z. Zheng, and N. B. Shroff, "Maximizing information in unreliable sensor networks under deadline and energy constraints," *IEEE Trans. Autom. Control*, vol. 58, no. 6, pp. 1416–1429, Jun. 2013.

[58] B. Alinia, H. Yousefi, M. S. Talebi, and A. Khonsari, "Maximizing quality of aggregation in delay-constrained wireless sensor networks," *IEEE Commun. Lett.*, vol. 17, no. 11, pp. 2084–2087, Nov. 2013.

[59] B. Alinia, M. H. Hajiesmaili, and A. Khonsari, "On the construction of maximum-quality aggregation trees in deadline-constrained WSNs," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 226–234.

[60] H. Yousefi, M. M. Koushki, B. Alinia, and K. G. Shin, "Maximizing quality of aggregation in WSNs under deadline and interference constraints," in *Proc. 15th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2018, pp. 1–9.

[61] Z. Zheng and N. B. Shroff, "Submodular utility maximization for deadline constrained data collection in sensor networks," *IEEE Trans. Autom. Control*, vol. 59, no. 9, pp. 2400–2412, Sep. 2014.

[62] M. Gary and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, 1st ed. New York, NY, USA: W. H. Freeman, 1979.

[63] M. Chen, S. Chang Liew, Z. Shao, and C. Kai, "Markov approximation for combinatorial network optimization," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6301–6327, Oct. 2013.

[64] A. Ghosh, Ö. D. Incel, V. S. A. Kumar, and B. Krishnamachari, "Multichannel scheduling and spanning trees: Throughput–delay tradeoff for fast data collection in sensor networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 6, pp. 1731–1744, Dec. 2011.

[65] O. Durmaz Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, "Fast data collection in tree-based wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 1, pp. 86–99, Jan. 2012.

[66] S. Ji, R. Beyah, and Z. Cai, "Snapshot and continuous data collection in probabilistic wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 3, pp. 626–637, Mar. 2014.

[67] F. Ghods, H. Yousefi, A. M. Afshin Hemmatyar, and A. Movaghar, "MC-MLAS: Multi-channel minimum latency aggregation scheduling in wireless sensor networks," *Comput. Netw.*, vol. 57, no. 18, pp. 3812–3825, Dec. 2013.

[68] B. Hohlt, L. Doherty, and E. Brewer, "Flexible power scheduling for sensor networks," in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw. IPSN*, 2004, pp. 205–214.

[69] H. Wu, Q. Luo, and W. Xue, "Distributed cross-layer scheduling for in-network sensor query processing," in *Proc. 4th Annu. IEEE Int. Conf. Pervas. Comput. Commun. (PERCOM)*, 2006, pp. 180–189.

[70] K. Kalpakis, K. Dasgupta, and P. Namjoshi, "Maximum lifetime data gathering and aggregation in wireless sensor networks," in *Proc. Netw.*, Aug. 2002, pp. 685–696.

[71] K. Dasgupta, K. Kalpakis, and P. Namjoshi, "An efficient clustering-based heuristic for data gathering and aggregation in sensor networks," in *Proc. IEEE Wireless Commun. Netw. WCNC*, Mar. 2003, pp. 1948–1953.

[72] H. Lee and A. Keshavarzian, "Towards energy-optimal and reliable data collection via collision-free scheduling in wireless sensor networks," in *Proc. IEEE INFOCOM 27th Conf. Comput. Commun.*, Apr. 2008, pp. 2029–2037.

[73] H. V. Luu and X. Tang, "Constructing rings overlay for robust data collection in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 36, no. 5, pp. 1372–1386, Sep. 2013.

[74] H. Van Luu and X. Tang, "An efficient algorithm for scheduling sensor data collection through multi-path routing structures," *J. Netw. Comput. Appl.*, vol. 38, pp. 150–162, Feb. 2014.

[75] J. Kirton, M. Bradbury, and A. Jhumka, "Source location privacy-aware data aggregation scheduling for wireless sensor networks," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 2200–2205.

[76] G.-S. Ahn, S. G. Hong, E. Miluzzo, A. T. Campbell, and F. Cuomo, "Funneling-MAC: A localized, sink-oriented MAC for boosting fidelity in sensor networks," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst. SenSys*, 2006, pp. 293–306.

[77] W.-Z. Song, R. Huang, B. Shirazi, and R. LaHusen, "TreeMAC: Localized TDMA MAC protocol for real-time high-data-rate sensor networks," *Pervas. Mobile Comput.*, vol. 5, no. 6, pp. 750–765, Dec. 2009.

[78] W.-Z. Song, F. Yuan, and R. LaHusen, "Time-optimum packet scheduling for Many-to-One routing in wireless sensor networks," in *Proc. IEEE Int. Conf. Mobile Ad Hoc Sensor Syst.*, Oct. 2006, pp. 81–90.

[79] H. Choi, J. Wang, and E. A. Hughes, "Scheduling for information gathering on sensor network," *Wireless Netw.*, vol. 15, no. 1, pp. 127–140, Jan. 2009.

[80] C. Florens and R. McEliece, "Packet distribution algorithms for sensor networks," in *Proc. IEEE INFOCOM*, Mar. 2003, pp. 1063–1072.

[81] C. Florens, M. Franceschetti, and R. J. McEliece, "Lower bounds on data collection time in sensory networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 6, pp. 1110–1120, Aug. 2004.

[82] C. Florens and R. McEliece, "Scheduling algorithms for wireless ad-hoc sensor networks," in *Proc. Global Telecommun. Conf., GLOBECOM*, Nov. 2002, pp. 6–10.

[83] S. Gandham, Y. Zhang, and Q. Huang, "Distributed minimal time convergecast scheduling in wireless sensor networks," in *Proc. 26th IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2006, p. 50.

[84] Y. Zhang, S. Gandham, and Q. Huang, "Distributed minimal time convergecast scheduling for small or sparse data sources," in *Proc. 28th IEEE Int. Real-Time Syst. Symp. (RTSS)*, Dec. 2007, pp. 301–310.

[85] S. Ergen and P. Varaiya, "TDMA scheduling algorithms for wireless sensor networks," *Wireless Netw.*, vol. 16, no. 4, pp. 1–13, 2009.

[86] S. C. Ergen and P. Varaiya, "PEDAMACS: Power efficient and delay aware medium access protocol for sensor networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 7, pp. 920–930, Jul. 2006.

[87] R. Ramaswami and K. K. Parhi, "Distributed scheduling of broadcasts in a radio network," in *Proc. IEEE INFOCOM 8th Annu. Joint Conf. IEEE Comput. Commun. Societies*, Apr. 1989, pp. 497–504.

[88] L. Paradis and Q. Han, "TIGRA: Timely sensor data collection using distributed graph coloring," in *Proc. 6th Annu. IEEE Int. Conf. Pervas. Comput. Commun. (PerCom)*, Mar. 2008, pp. 264–268.

[89] Z. Shen, H. Jiang, and Z. Yan, "Fast data collection in linear duty-cycled wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 4, pp. 1951–1957, May 2014.

[90] H. Lee, A. Keshavarzian, and H. Aghajan, "Near-lifetime-optimal data collection in wireless sensor networks via spatio-temporal load balancing," *ACM Trans. Sensor Netw.*, vol. 6, no. 3, pp. 1–32, Jun. 2010.

[91] R. Zhang, X. Cheng, X. Cheng, and L. Yang, "Interference-free graph based TDMA protocol for underwater acoustic sensor networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4008–4019, May 2018.

[92] Z. Liao, D. Li, and J. Chen, "Joint bandwidth optimization and media access control for multihop underwater acoustic sensor networks," *IEEE Sensors J.*, vol. 15, no. 8, pp. 4292–4304, Aug. 2015.

[93] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient collision-free medium access control for wireless sensor networks," in *Proc. 1st Int. Conf. Embedded Netw. Sensor Syst. SenSys*, 2003, pp. 181–192.

[94] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu, "Z-MAC: A hybrid MAC for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 511–524, Jun. 2008.

[95] I. Rhee, A. Warrier, J. Min, and L. Xu, "DRAND: Distributed randomized TDMA scheduling for wireless ad-hoc networks," in *Proc. 7th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. MobiHoc*, 2006, pp. 190–201.

[96] S. Gobriel, D. Mosse, and R. Cleric, "TDMA-ASAP: Sensor network TDMA scheduling with adaptive slot-stealing and parallelism," in *Proc. 29th IEEE Int. Conf. Distrib. Comput. Syst.*, Jun. 2009, pp. 458–465.

[97] O. Chipara, C. Lu, and J. Stankovic, "Dynamic conflict-free query scheduling for wireless sensor networks," in *Proc. IEEE Int. Conf. Netw. Protocols*, Nov. 2006, pp. 321–331.

[98] W. Zhao and X. Tang, "Scheduling sensor data collection with dynamic traffic patterns," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 4, pp. 789–802, Apr. 2013.

[99] W. Zhao, X. Tang, and L. Xu, "Constructing routing structures for sensor data collection with dynamic traffic patterns," *Comput. Netw.*, vol. 142, pp. 272–290, Sep. 2018.

WENBO ZHAO (Member, IEEE) received the B.Sc. degree from the Computer Science Department, Nanjing University of Astronautics and Aeronautics, China, in 2006, and the Ph.D. degree in computer engineering from Nanyang Technological University, in 2014. She is currently a Lecturer with the School of Aerospace Science and Technology, Xidian University, China. Her research interests include cyber physical systems, the Internet of Things, and wireless and mobile networking.

YIFAN LI received the B.E. degree from the Department of Electronic and Information Engineering, Huazhong University of Science and Technology, China, in 2008, and the Ph.D. degree from the School of Computer Engineering, Nanyang Technological University, in 2014. She worked as an Engineer with China Railway Siyuan Survey and Design Group Company Ltd. She is currently a Lecturer with the School of Logistics Engineering, Wuhan University of Technology. Her research interests include cooperation aided-transmission in mobile wireless networks, power management in smart grid networks, and resource allocation for M2M communications in LTE networks.

BO YAN received the B.S. degree in communication engineering from Northwest University, China, in 2013, and the Ph.D. degree in guidance navigation and control from Xidian University, in 2018. He is currently with Xidian University. His research interests include radar data prsocessing, multi-target detection and tracking, and data fusion.

LUPING XU received the Ph.D. degree in signal and information processing from Xidian University, in 1996. Since 2000, he has been a Professor with Xidian University. His main research interests include target detection, spread spectrum, satellite and mobile communication, and SAR image processing.