

Communication Synchronization in Cluster-Based Sensor Networks for Cyber-Physical Systems

XI DENG AND YUANYUAN YANG

Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, USA

CORRESPONDING AUTHOR: X. DENG (xi.deng@stonybrook.edu)

This work was supported in part by the U.S. NSF under Grant ECCS-0801438 and U.S. ARO under Grant W911NF-09-1-0154.

ABSTRACT A reliable, scalable, and low-delay information collection network is an essential component in cyber-physical systems (CPS). Cluster-based sensor network is a good candidate because of its advantage in increasing scalability, improving energy efficiency, and providing QoS guarantees. However, in such networks, frequent interactions between the intra-cluster communication and the inter-cluster communication are inevitable, which may severely downgrade the communication efficiency and hence the network performance if not handled properly. Proper synchronization among these two types of communications is required. In this paper, we propose two approaches to schedule the communications in clustered wireless sensor networks aiming at delay-sensitive applications. In the first approach, an efficient cycle-based synchronous scheduling is proposed to achieve low average packet delay and high throughput by optimizing the cycle length and transmission order. In the second approach, a novel clustering structure is introduced to eliminate the necessity of communication synchronization so that packets are transmitted with no synchronization delay, yielding very low end-to-end packet delay. Our extensive experimental results demonstrate the superior performance of both the approaches. These two approaches are then integrated as a hybrid scheme that allows smooth switching between them. The hybrid scheme takes advantage of both the approaches and enables cluster-based sensor networks to serve as the fundamental network infrastructure for information collection in CPS.

INDEX TERMS Clustering, wireless sensor networks, communication synchronization, delay-sensitive, data gathering.

I. INTRODUCTION

The emerging Cyber-Physical System (CPS) has been gradually changing the society and the world by interacting with people's everyday life. Its research and use can be found in various applications of different societal services, including efficient energy control systems, intelligent traffic monitoring, medical care system and etc. [1]. In general CPS, information of physical world is collected and analyzed for the computing system to make appropriate decisions and controls responding to any physical situations and changes. While there are design challenges in different aspects of CPS, we focus on providing a fundamental infrastructure for information collection in CPS.

Wireless sensor network (WSN), as originally designed to perform sensing tasks, is a natural fit for information collection in CPS. Compared to conventional WSNs, WSNs adopted in CPS faces more stringent design challenges.

- WSNs should have good scalability to adapt to the increasing geographical range covered by CPS.
- WSNs should be performed in a low-latency fashion to support real-time interaction with the physical world.
- WSNs should still be energy efficient to support long and stable service for CPS. Notice that previous energy efficient solutions in WSNs may not be suitable in CPS as they may not satisfy the other two requirements simultaneously.

Of all kinds of topologies in WSNs, clustering is a good candidate to meet the above challenges, considering its wide use in WSNs to increase scalability, improve energy efficiency and provide QoS guarantees. With clustering, sensor nodes are organized into clusters and a cluster head (CH) node is selected for each cluster according to certain rules, while other nodes act as members in the clusters. In

cluster-based data gathering, data collected by cluster members are first sent to CHs, which in turn deliver the data to the data sink either by direct communication or through relays on intermediate CHs. While clustering is initially introduced to achieve energy efficiency, it can also help maintain low packet latency in delay-sensitive data gathering. This is because that packets from different members can be combined as aggregated packets at CHs to reduce the transmission overhead of packet headers and control packets (e.g., ACK packets), leading to shortened transmission delay and increased energy efficiency. In addition, clustering simplifies the routing from the source node to the sink, and shorter routing paths reduce network traffic as well.

To support low-latency data gathering, however, cluster-based WSNs encounter a new communication synchronization problem due to their more complex communication patterns compared to WSNs with a flat topology. In general, the communication in a cluster-based WSN includes intra-cluster communication among sensors in the same cluster and inter-cluster communication among different CHs. Intra-cluster communication in each cluster is usually controlled by the CH with a Time-Division-Multiple-Access (TDMA) based protocol to avoid transmission collisions. For inter-cluster communication, CHs can be considered to form a smaller *relay network* where either TDMA or Carrier-Sense-Multiple-Access (CSMA) based protocols can be utilized. To avoid interference between intra- and inter-cluster communications, different channels are used for two types of communications, which implies that CHs have to switch between two channels accordingly as most of sensors can operate on only one radio channel at a time. Let *i-state* and *o-state* denote that a CH uses the channel for intra-cluster communication and inter-cluster communication, respectively. Such state switching thus incurs a synchronization problem which is critical for delay-sensitive applications: the sending CH and the receiving CH should be in *o-state* simultaneously and any inter-cluster packet transmission may endure an unacceptable long delay before the receiver switches to *o-state*. In such a case, the inter-cluster packet that consists of multiple sensing packets may become useless and be discarded, causing severe performance loss.

Previous work that targets at the energy efficiency of clustering handles this synchronization problem by simply grouping the intra- and inter-cluster communications involved in all clusters into two global and non-overlapping periods. In this approach, since the intra- and inter-cluster communications are guaranteed to be performed separately, the synchronization problem can be avoided. However, by intentionally separating two types of communications, such an approach may cause low channel utilization and hence long end-to-end delays, rendering it not suitable for delay-sensitive applications. In this paper, we first propose two communication scheduling approaches to solving the synchronization problem from different angles and support delay-sensitive data gathering applications with different requirements.

We then discuss the integration of the two approaches and its suitability to use in CPS.

We first propose a TDMA based, synchronous scheduling approach to achieve low end-to-end packet delay by converting the cluster synchronization problem into a scheduling problem in generic wireless networks. Due to the NP-completeness of the scheduling problem, we propose an efficient heuristic scheduling algorithm. Compared to other cycle-based approaches in the literature, our approach owns three unique features. First, the cluster heads can individually decide their intra-cluster packet collection time, rather than a globally synchronized collection time. Second, since all packets are sent to the sink, we schedule transmissions according to the order of nodes in the routing path to minimize the queueing delay. Third, we efficiently overlap the transmissions so that the cycle length and the average end-to-end packet delay can be reduced as much as possible. Experiments show our approach can achieve 50% shorter average packet delay than the existing approach.

The cycle based approach may have some restrictions in a harsh environment due to its vulnerability to the clock drift, topology changes and irregular interferences occurred in WSNs [18]. We thus propose a CSMA based approach that solves the communication synchronization problem asynchronously. The approach is constructed on a new clustering structure with a new type of node, called *relay node*, rather than the conventional CH-member structure. The relay nodes stay in *o-state* and replace the CHs to receive and forward the aggregated packets. With the assistance of relay nodes, inter-cluster communications are automatically synchronized. Compared to the first approach, the asynchronous approach better utilizes the wireless channel and yields even lower packet delay. The performance of both approaches has been verified through extensive ns-2 simulations.

While these two approaches can fulfill different performance requirements, they are then integrated as a hybrid scheduling scheme that fully exploits the benefit from both approaches and allows dynamic switching among them to adapt to various network conditions. When emergency occurs or wireless signal quality is poor, the second approach is preferred to enjoy lower latency and higher interference tolerance; when the emergency ends, the first approach can be used to achieve higher energy efficiency based on the nature of TDMA. We believe the hybrid approach is suitable to support the fundamental network for information collection in CPS.

The rest of the paper is organized as follows. Section II discusses the related work. Section III and Section IV describe the design of synchronous and asynchronous approaches, respectively. The integration of the two approaches is then elaborated in Section V. Section VI presents the experimental results for the proposed approaches. Finally, Section VII gives the conclusion.

II. RELATED WORK

Clustering is a popular topology control approach to achieving energy efficiency and scalability in WSNs. In this section,

we briefly review the cluster formation algorithms and then discuss some existing work concerning communication protocols in cluster-based networks.

Cluster formation algorithms have been extensively studied in the literature. Their primary purpose is to consider load balance and energy efficiency to prolong network lifetime. While some algorithms consider a heterogeneous environment where CHs are more powerful than regular sensor nodes [2], [5], [6], other algorithms consider a homogeneous environment where CHs are ordinary sensors [7]. In this paper, we mainly focus on clustering in a homogeneous environment. Typical algorithms in this category include LEACH [3] and HEED [4]. LEACH selects CHs randomly and distributes energy consumption evenly among all nodes by cluster head rotation. HEED selects CHs by considering the residual energy in the nodes and the communication cost. A comprehensive survey on different clustering algorithms can be found in [7].

There is also some work on communication protocols in cluster-based networks. While intra-cluster communication in these protocols is always TDMA-based, the inter-cluster communication adopts different approaches in different works. In [8], a round-based data collection scheme with direct sink access was proposed. It was assumed that CHs can directly access the sink, which has the capability of multiple packet reception. The intra- and inter-cluster packet delay was considered separately and the end-to-end delay was not studied. A similar round-based protocol was proposed in [9], where the routing path for inter-cluster communication consists of either cluster heads or a combination of CHs and members. In [10], a pure TDMA-based scheme was proposed to achieve optimized energy efficiency and minimum delay, in which the packet delay is directly associated with the length of TDMA frame. In addition, the MAC protocol defined in IEEE 802.15.4 can be utilized in cluster-based networks [11]. In particular, a cluster-tree topology is constructed with each CH corresponding to a coordinator, which maintains a superframe with 16 slots. The members are allowed to communicate with the CH in any slot in the superframe. In general, superframes for different coordinators do not overlap so that the interference among different clusters can be avoided.

In the above protocols, communication synchronization is handled by either setting a complete transmission schedule for every CH, or globally separating the intra- and inter-cluster communications. As will be seen in the performance evaluation section later, compared to our proposed approaches, these approaches do not perform well in delay-sensitive data gathering.

Finally, many communication protocols in cluster-based networks adopt hybrid approaches that utilize both TDMA and CSMA. Such hybrid approaches are also commonly seen in general WSNs, such as S-MAC [12], T-MAC [13], Z-MAC [23] and funneling-MAC [24]. S-MAC maintains continuous duty cycles and employs CSMA in each cycle for transmissions. T-MAC follows a similar hybrid approach and improves S-MAC in terms of energy consumption by using a

listening window at the beginning of each cycle. Z-MAC tries to pertain the advantage of both TDMA and CSMA such that the hybrid approach acts like CSMA under light traffic and TDMA when traffic becomes heavier. This goal is achieved by assigning ownership to each time slot and giving the owner higher priority to access the channel. Funneling-MAC also utilizes the hybrid approach to solve the funneling problem, i.e., nodes closer to the sink have much heavier traffic and incur more communication control overhead. Thus, TDMA is used for these nodes to avoid frequent contentions while CSMA is used for other nodes with less contentions. These protocols, however, are designed for general WSNs and cannot be directly applied in cluster-based WSNs.

III. SYNCHRONOUS SCHEDULING

In this section, we present Cycle-Based Scheduling (CBS), a TDMA based, synchronous scheduling approach. To begin with, we describe the assumptions for the system, which are also shared by the asynchronous scheduling approach.

A. ASSUMPTIONS

The network considered is a WSN with n sensor nodes randomly distributed in a 2D region. We consider a typical data gathering application in WSNs where all sensor nodes send collected data to a single sink. We also make the following assumptions on the WSN.

- The clustering topology is pre-constructed by a clustering algorithm, such as the algorithms mentioned in Section II, which indicates that the size of the different clusters may be different. In addition, we assume the topology of the relay network is stable during the data gathering. This is reasonable if the CHs are properly selected with adequate energy.
- Sensors can transmit on different radio channels. However, they can only transmit or receive packets on one channel in any instant. Different radio channels do not interfere with each other.
- Sensors have the same sensing rate λ and sensing packets are of the same length. The packet generation process is assumed to be Poisson.

Under the above assumptions, which are applicable in many real-world networks, next we describe the details of CBS.

B. BASIC COMMUNICATION CYCLE

CBS schedules communications in consecutive cycles and each node is assigned some fixed conflict-free intervals to transmit and receive packets in each cycle. Nodes only wake up in the assigned intervals and sleep otherwise to reduce energy consumption. Each node is assigned a single interval for transmission so that the synchronization overhead between the transmission pair is minimized. The goal of the scheduling is to minimize the average end-to-end packet delay. We consider this problem by separating the intra- and the inter-cluster communications.

1) INTRA-CLUSTER COMMUNICATION

Intra-cluster communication includes all transmissions from cluster members to the CH. Since interferences from other clusters can be avoided by assigning different radio channels to adjacent clusters, the communications within a cluster are independent and hence it is reasonable to only consider a single cluster.

We limit all communications for a cycle in a consecutive period so that the CH needs not to frequently switch between intra- and inter-cluster communications. As will be seen later, the duration of this period is relatively short compared to the cycle length, we simply consider a general TDMA scheme for the intra-cluster period. The whole period is divided into multiple identical time slots whose length τ is set equal to the time required for a packet transmission. Packets are sent in these time slots directly from cluster members to the CH. Each node is assigned the same k time slots given their same packet generation rate. For simplicity, we assume the CH is also assigned k time slots for necessary control packets. Assume the cluster has m nodes, the duration of the intra-cluster period is thus $m \cdot k \cdot \tau$.

For such scheduling, we are concerned with the determination of k and the packet collection delay, which is defined as the elapsed time between the packet is generated and the end of the intra-cluster period in which the packet is collected.

Lemma 1: The lower bound of k is $\lceil \lambda T \rceil$, where T is the cycle length.

Proof: The expected number of packets generated by each node in a cycle is λT . Since a cluster member can transmit one packet in a time slot, in order to collect all packets in one intra-cluster period, it must satisfy $k \geq \lambda T$. Since k can only be an integer, the lower bound of k is $\lceil \lambda T \rceil$.

Lemma 2: If k is large enough for collecting all packets in a cycle, the expected collection delay is $\frac{T+m \cdot k \cdot \tau}{2}$.

Proof: Consider a packet generated by node i , whose time slots assigned end at s_i . Since k is large enough for collecting all packets in a cycle, this packet must be generated between the end of slot s_i of two consecutive intra-cluster periods and this interval is T . Since the packet generation process is Poisson, the time a particular packet is generated within a fixed interval is uniform [19], thus the expected generation time in this interval is $\frac{T}{2}$ and the expected collection time is $\frac{T}{2} + (m \cdot k - s_i)\tau$. Therefore, the expected collection time for all packets will be

$$\begin{aligned} E(D_c) &= \frac{T}{2} + m \cdot k \cdot \tau - E(s_i) \\ &= \frac{T + m \cdot k \cdot \tau}{2}. \end{aligned} \quad (1)$$

Lemma III-B.1 indicates that the intra-cluster period can be placed at any position in the cycle without affecting the collection delay, which is only dependent on the cycle length and the period duration. It also suggests that k can be selected

at its lower bound $\lceil \lambda T \rceil$ to minimize the collection delay, which monotonically increases with k .

2) INTER-CLUSTER COMMUNICATION

Inter-cluster communication includes transmissions in the relay network, which consists of CHs and the sink. For data gathering, as the relay network is stable, the CHs are organized into a fixed routing tree rooted at the sink at the same time when the clusters are formed. The construction of the routing tree is independent of our scheduling approach and thus is not discussed in this paper. Within a cycle, each CH is assigned an interval to send all packets, including packets collected by itself and packets received from other CHs, to its parent. The practical length of this interval should be slightly longer than the transmission time of all packets to accommodate the necessary control packets such as ACK and potential synchronization errors. However, since we are focusing on the cycle scheduling, we set the length equal to the transmission time of all packets for simplicity.

The relay network can be viewed as a general wireless network except that the CH is not available during intra-cluster period. We introduce an *intra node* for each CH to represent the intra-cluster packet collection. Intra node i generates $m_i \cdot k$ packets in each cycle with zero queueing delay before the packets are sent out. It transmits packets to the CH within an interval whose duration equals $m_i \cdot k \cdot \tau$. The transmission does not affect other nodes except for the associated CH. Thus the considered problem becomes to schedule intervals for all nodes in the transformed network.

C. INTERVAL SCHEDULE

To obtain an efficient interval schedule, we first present an analytical model for the problem and then show an illustration example. Guided by the example, we will propose our scheduling approach.

1) MATHEMATICAL MODEL

The network is represented by a graph $G = (V, E)$. V is the set of nodes, including the sink, the CHs and the corresponding intra nodes. Denote p_i as the parent of node i in the routing tree and (i, j) as a transmission link between node i and node j , then $E = \{(i, p_i) | i \in V\}$. Since every node has a fixed parent, it is easy to see $|V| = |E| + 1$.

To model the interference of transmissions, we construct a conflict graph $G' = (V', E')$. V' represents all the transmission links in E . For simplicity, we use i to represent link (i, p_i) such that $V' = V \setminus \{v_s\}$, where v_s represents the sink. E' is constructed such that if $(i, j) \in E'$, nodes i and j cannot transmit at the same time due to that the distance between any two of nodes i, j, p_i and p_j is within the transmission radius. Such construction is valid if we assume that the receiver may send ACK packets and transmissions will not only be affected by the sender, but also by the receiver. For an intra node i , there is only one conflict edge (i, p_i) corresponding to the fact that the CH cannot send packets during the transmission of its corresponding intra node.

The interval scheduling problem is to find a feasible time interval (s_i, f_i) for each node i in V' , where s_i and f_i are the starting and finishing time instant with $0 \leq s_i \leq f_i$. The cycle length is then set as $t = \max_{i \in V'} f_i$. Here we normalize the cycle to time slots with length $k \cdot \tau$ so that the actual cycle length $T = k \cdot \tau \cdot t$. Since the interval equals the transmission time of all packets, we have $f_i = s_i + n_i$, where n_i is the maximum number of packets node i sends in a cycle and can be obtained by

$$n_i = \begin{cases} m_i, & \text{if } i \text{ is an intra node} \\ \sum_{j \in C_i} n_j, & \text{if otherwise.} \end{cases}$$

Here, C_i denotes the child set of node i . For an interval of node i to be feasible, its transmission link should not conflict with any other transmission links, thus $\forall (i, j) \in E', s_i \geq f_j$ or $s_j \geq f_i$.

The end-to-end packet delay can be broken down into transmission delay and queueing delay. The transmission delay from an intra node to the corresponding CH is simply the collection delay in the intra-cluster period while the transmission delay from CH i to its parent p_i is n_i . The queueing delay, defined as the waiting time of a packet at a node before it is sent out, will be $(s_{p_i} - f_i + t) \bmod t$ for a parent p_i . Thus the average packet delay is

$$D = k \cdot \tau \cdot \frac{\sum_{i \in V'} n_i (d_i + (s_{p_i} - f_i + t) \bmod t)}{n_{v_s}} \quad (2)$$

where

$$d_i = \begin{cases} \frac{t+m}{2}, & \text{if } i \text{ is an intra node} \\ n_i, & \text{if otherwise.} \end{cases}$$

The optimal scheduling problem was proved to be NP-complete by reducing the K-Colorability problem to the scheduling problem [20]. Thus, we will design a heuristic algorithm for the problem. Before that, we examine an example to reveal some interesting property in the scheduling problem.

2) EXAMPLE OF CHAIN TOPOLOGY

This example considers a network with chain topology as shown in Fig. 1(a). Each CH has a corresponding intra node, which generates one packet in a cycle. Thus, node i will send i packets in a cycle.

Fig. 1(b) shows an intuitive scheduling, where node i is sequentially assigned an interval of i and all intra nodes are assigned an interval of 1 at the beginning of the cycle. This is actually a not-so-bad scheduling as it eliminates the queueing delay in the relaying: a CH will immediately send out all the packets upon its receiving from its child. The average packet delay $D = 32.5$.

The intuitive scheduling does not consider the fact that packets have zero queueing delay on the intra node. An improved scheduling in Fig. 1(c) schedules the intervals for intra nodes as close as possible to the corresponding CHs. The average packet delay $D = 29.2$.

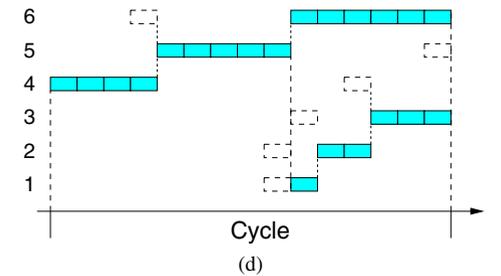
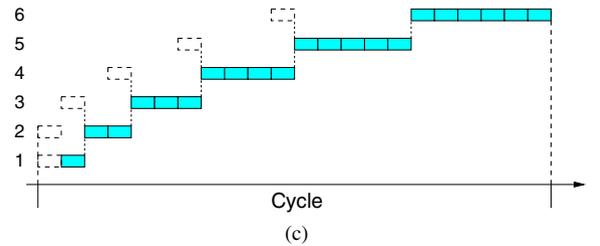
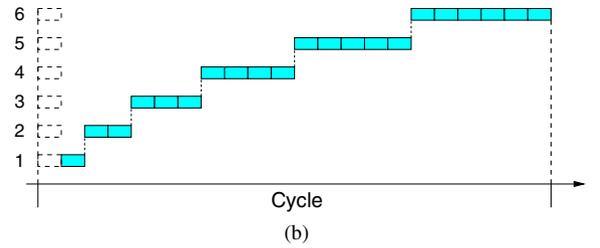
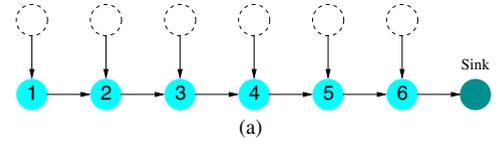


FIGURE 1. An example network with chain topology.

In the improved scheduling, the collection delay on intra node, which is 11.5 according to Eq. (1), takes a large part in the total delay. Since the delay mainly depends on the cycle length, we can further reduce the average packet delay by reducing the cycle length. Fig. 1(d) shows the optimal scheduling that minimizes the cycle length. In the scheduling, the cycle length is exactly the sum of the interval lengths of nodes 4, 5 and 6. Since these nodes are conflicting with each other, their intervals cannot overlap and thus the cycle length reaches its minimum. On the other hand, the intervals of nodes 1, 2 and 3 are placed at the end of the cycle so that no extra queueing delay is introduced. The average packet delay $D = 25.7$.

Following this example, we obtain three guidelines to design the scheduling algorithm.

- Interval assignment should follow the order of the nodes in the routing path.
- Intervals for intra nodes should be as close as possible to the intervals for the corresponding CH.
- The cycle length can be reduced by overlapping as much intervals as possible. However, the reduction is bounded

by the intervals that cannot be overlapped due to the conflicts. In fact, although each interval will have a number of conflicted intervals, it is the longer intervals that decide the lower bound of the cycle length.

3) ALGORITHM

The algorithm is summarized in Tables 1 and 2. Table 1 describes a basic scheduling algorithm that strictly schedules the nodes according to their orders in the routing path: a node is always scheduled before its parent. From the second guideline, we see that nodes should be scheduled as late as possible. Thus, the algorithm actually schedules nodes in the reverse order so that nodes can be scheduled closer to the end of the cycle. For illustration, we define function $I(\cdot)$ on nodes such that

$$I(i) = \begin{cases} 1, & \text{if } i \text{ is scheduled} \\ 0, & \text{if } i \text{ is not scheduled.} \end{cases}$$

During the scheduling, a set V_t is maintained to include nodes whose parents are already scheduled. The algorithm tries to find the earliest possible scheduling interval for all nodes in V_t and the node with the earliest starting time is scheduled. Then the algorithm schedules the next node with no earlier starting time until all nodes are scheduled or no intervals can be scheduled within the required range. Then the reverse schedule is obtained.

TABLE 1. Basic Scheduling Algorithm.

<p>Input: node set to be scheduled V_s, schedule range (S, F) Output: updated schedule $V_t = \{i i \in V_s, I(i) = 0, I(p_i) = 1\}$ while $V_t \neq \emptyset$ order V_t by distance to sink find earliest schedulable interval $(s_i, f_i), s_i \geq S$ for each node in V_t. $j = \arg \min_{i \in V_t} s_i$ if $f_j > F$ return schedule node j with (s_j, f_j) $S = s_j$ update V_t end while</p>
--

Table 2 utilizes this basic algorithm to perform the actual scheduling. The idea is to first determine a tentative cycle length and then try to schedule all the intervals within this cycle. Given the third guideline, we start to schedule the intervals from the nodes that are closer to the sink. For assistance, we construct two node sets V_n and V_c . Let

$$V_n = \{i | I(i) = 0, I(p_i) = 1, i \in V'\}.$$

Initially we assume $I(v_s) = 1$ so that V_n includes all nodes that directly send packets to the sink. Clearly, these intervals cannot be overlapped. In addition, their conflicting intervals cannot be overlapped with these intervals either. For that, we construct

$$V_c = V_n \cup \{j | (i, j) \in E', I(j) = 0, i \in V_n\}.$$

We then schedule V_c with the basic scheduling algorithm with no range requirement and obtain the tentative cycle length. For other nodes that are not scheduled, since they are not in the current V_c , it is guaranteed that their scheduled intervals can be overlapped with intervals for nodes in V_n . Thus we can schedule the rest of nodes from the beginning of the cycle. Thus we update V_n and V_c according to current schedule and repeat the basic scheduling algorithm. Since nodes that are closer to the sink have longer intervals, in most cases the updated V_c can be scheduled at the beginning part of the cycle, leaving the rest part of the cycle available for further scheduling. We then schedule the rest of nodes to fill in the available part of the cycle to avoid queueing delays. This process is repeated until all nodes are scheduled. The finiteness of this process is guaranteed by the construction of V_n , which guarantees that all children of already scheduled nodes will be scheduled in the next iteration. In fact, our experiments show that two iterations will suffice in most cases as the tentative cycle length is large enough for the rest of nodes to schedule sequentially. Notice that since the basic scheduling algorithm schedules nodes reversely, the actual schedule should be in the exactly reverse order of the obtained schedule.

TABLE 2. Actual Scheduling Algorithm.

<p>Input: graph $G = (V, E)$ and conflict graph $G' = (V', E')$. Output: interval schedules for nodes in V' $t = 0$ construct V_n, V_c while $V_c \neq \emptyset$ schedule V_c with range $[0, \infty)$ $t_c = \max_{i \in V_c} f_i$ if $t_c < t$ $V_u = \{i i \in V', I(i) = 0\}$ schedule V_u with range $[t_c, t]$ end if $t = \max(t, t_c)$ update V_n, V_c end while reverse the whole schedule</p>
--

4) ANALYSIS

With this cycle-based scheduling, we are also interested in determining the maximum packet generation rate. Recall $k \geq \lambda T$. We have

$$k \geq \lambda T = k \cdot \lambda \tau t,$$

$$\lambda \leq \frac{1}{\tau t}.$$

Therefore, the maximum packet generation rate is $\frac{1}{\tau t}$. On the other hand, when the generation rate does not exceed the maximum rate, it is always satisfied that $k \geq \lambda T$. Thus, we can always set $k = 1$ to minimize the packet delay.

IV. ASYNCHRONOUS SCHEDULING APPROACH

In this section, we present the second scheduling approach, which is called New Cluster Scheduling (NCS), adopts an

asynchronous approach that essentially avoids the synchronization problem by introducing a new clustering structure. Next we first introduce the new clustering structure and then describe the approach in detail.

A. NEW CLUSTERING STRUCTURE

Instead of designing another algorithm to globally schedule all the inter-cluster communications for synchronization, NCS attempts to simplify the synchronization by changing the communication pattern. To achieve this goal, NCS introduces a new clustering structure, which includes a new type of node: relay node.

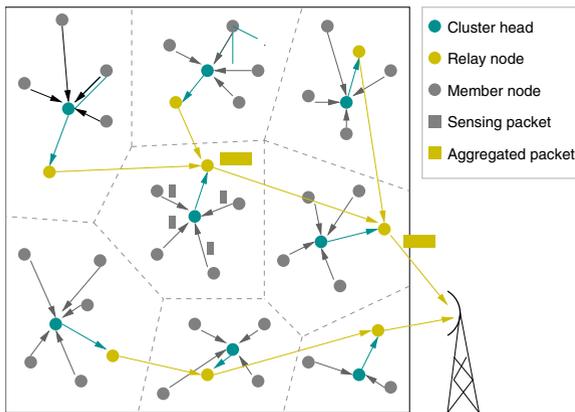


FIGURE 2. The new clustering structure includes CHs, relay nodes and members. The packet transmissions for the center cluster are shown.

The new clustering structure is illustrated in Fig. 2, in which a cluster contains a CH node, a relay node and multiple cluster members. The relay nodes always stay in o-state and only participate in inter-cluster communications. During data gathering, while cluster members still send sensing packets to the corresponding CH, the CH no longer sends the aggregated packet to the next-hop CH but sends to the relay node of its own cluster instead. Upon receiving the packets, the relay node further combines them with its own sensing packets and forwards the packets to the next-hop relay node until the packets reach the sink. With such communication pattern, the communication synchronization is greatly simplified. CHs can continue intra-cluster data collection immediately after sending out the aggregated packet, reducing the data collection delay. In the meanwhile, inter-cluster communication can be performed without any restrictions, incurring no waiting delays for synchronization. The wireless channel thus can be better utilized and lower packet delay can be achieved.

On the other hand, the new clustering structure does not substantially increase the complexity of the cluster formation process. A network with the new clustering structure can be simply converted from a network with the conventional clustering structure by selecting the member with the highest residual energy as the relay node in each cluster. The routing algorithms for creating routes among different CHs in

conventional cluster-based networks can also be utilized to create routes among CHs and relay nodes.

B. APPROACH DETAILS

NCS adopts the same TDMA protocol as CBS for intra-cluster communications and CSMA protocol for inter-cluster communications. While member nodes and relay nodes are fixed in i-state and o-state, respectively, CHs still need to switch between two states, which is the major task of NCS. Since there is no synchronization required among different CHs, the state switching, or the duration at each state, can be determined independently for each CH.

We first determine the inter-cluster duration a CH stays in o-state. When a CH switches to o-state, it cannot transmit a packet immediately. Consider the case that CH 1 is sending a packet to CH 2 at o-state while CH 3, a neighbor of CH 2, switches to o-state. If CH 3 is not in the transmission range of CH 1, it will not detect the ongoing transmission, which would be interrupted by any transmission initiated at the CH before the end of the current transmission. In this case, the protection from RTS/CTS handshake fails as they were not received by the CH who was in i-state then. We call the period in which such collisions may occur the *blind* period and its duration equals the transmission time of a packet of a maximum allowable packet length. After this blind period, the CH sends the aggregated packet to the next-hop relay node on the routing path. Once the transmissions are completed, the CH can immediately switch back to i-state to continue data collection.

Next we consider the intra-cluster duration of a CH in i-state. Since the CH does not participate in inter-cluster communications for other CHs, the duration in i-state only affects its own collection delay. Intuitively, to minimize the collection delay, the CH can switch to o-state immediately after the end of the time frame in which a packet is collected. However, such an approach yields relatively small aggregated packets, which underutilizes the wireless channel due to the overhead of packet headers and control packets, lowering the maximum achievable throughput. Alternatively, we use a fixed collection duration, denoted as T_c . A larger T_c indicates less frequent data collection, yielding a smaller number of larger aggregated packets. Consequently, the channel is better utilized and higher throughput can be achieved. On the other hand, a larger T_c also leads to longer collection delay and hence the end-to-end packet delay. Therefore, adjusting T_c can obtain different tradeoffs between the packet delay and the maximum achievable throughput.

T_c determines the number of time slots in an intra-cluster period. Following a similar analysis to that in Section III-B, we see that the necessary number of time slots for a member in an intra-cluster period is $k = \lceil \lambda(T_c + T_o) \rceil$, where T_o represents the duration of the last inter-cluster period. When $T_c > m \cdot k \cdot \tau$, a portion of the intra-cluster period is actually wasted. For energy efficiency, we organize the intra-cluster period into time frames with each consisting m time slot, allowing each node to send a packet in a time frame.

Then the CH can remain active only in the last k frames and sleep in other times. The entire process for a CH is described in Fig. 3.

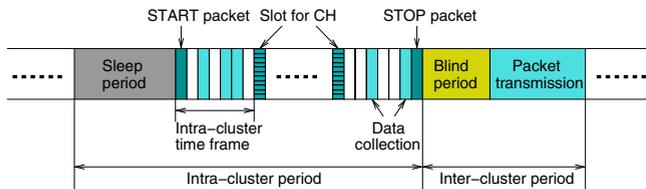


FIGURE 3. Timing of data gathering at a CH in NCS.

C. DELAY GUARANTEE

Thanks to the relay nodes, NCS avoids the synchronization delays during the inter-cluster communications, allowing the relay network to operate similarly to a general WSN. Therefore, although it does not directly provide delay guarantees, it greatly facilitates the utilization of real-time routing protocols at the upper layer, such as SPEED [21] and MMSPEED [22], which relies heavily on the one-hop packet delays. These delays could be very long and irregular in a cluster-based network where communications incur synchronization delays, degrading the performance of the real-time routing protocols. On the contrast, with NCS eliminating the synchronization delays, real-time routing protocols can be easily implemented to provide optimal performance on delay guarantees.

V. INTEGRATING CBS AND NCS

CBS and NCS are designed to satisfy different network requirements, with each having its own tradeoff. CBS is more energy efficient due to the nature of TDMA: nodes can sleep in any idle slots to preserve energy. On the other hand, NCS yields lower end-to-end packet delay as will be shown in the evaluation in Section VI. Moreover, the CSMA based protocol makes NCS more tolerant to interference and collisions. In a general CPS application, we envision that the targeted emergencies do not occur often. As a result, CBS will be sufficient to monitor the environment in most time, and the network can switch to NCS to track the emergency for prompt reaction. Thus, A hybrid scheduling scheme that integrates CBS and NCS can meet the CPS requirements of both achieving low latency when necessary and preserving a long life time.

The major challenge in the hybrid scheme is the switching between the adoptions of CBS and NCS. Intuitively, scheduling switching can be performed concurrently with the next round clustering which reconstructs clusters. However, such reconstruction is both energy and time consuming, causing interrupts on the current monitoring task as well. Therefore, the switching should be smoothly performed on the current clusters, which we discuss in this section.

A. SWITCHING FROM CBS TO NCS

The switching from CBS to NCS occurs when the emergency or severe channel interference is detected by the sink through

the analysis of the received data. This process can be divided into two major tasks: to adapt the current cluster structure to NCS while maintaining the connectivity, and to notify all clusters to perform the switching smoothly.

1) CLUSTER STRUCTURE ADAPTION

Switching to NCS requires a node in each cluster to be selected as the relay node. Such selection should be very carefully done since in NCS, the relay nodes are responsible for the connectivity of the relay network. In practice, we select the current CH as the relay node for each cluster so that the topology of the relay network keeps unchanged and the network naturally remains connected during the switching. Such selection also eliminates the potential updates of the inter-cluster routing information for all clusters, minimizing the inter-cluster communication overhead. In addition, maintaining a connected relay network during the switching facilitates the possible cluster reorganization: in each newly constructed cluster, the CH simply needs to find a relay node to keep the cluster connected. We do not further articulate the cluster reorganization process as it is beyond the scope of the paper.

With the CH becoming the relay node, we select the node with the highest residual energy from the remaining nodes as the new CH. A new intra-cluster scheduling can then be easily performed according to the specification of NCS.

2) SWITCHING NOTIFICATION

Switching is initiated by the sink, which spreads the switching notification in the reverse order of data collection: the sink and the CHs are responsible for notifying their direct children. The notification can be simply piggybacked in the ACK packet when a data packet is received from an un-notified child. The notified child can then switch to NCS accordingly and notify its own children in the next cycle until the switch is completed on the whole network.

B. SWITCHING FROM NCS TO CBS

When the sink decides that the network can return to regular monitoring, a switching from NCS to CBS is necessary to increase energy efficiency. Similarly, we also consider the two major tasks in the switching: cluster structure adaption and switching notification.

The more challenging task is the cluster structure adaption, which is essentially the CH selection problem for CBS. Letting the relay node or the current CH in NCS become the next CH in CBS can have certain advantages. Selecting the relay node can easily maintain the network connectivity while selecting the current CH can maintain the current TDMA scheduling with a slight modification of adding a slot for the relay node. However, both may not be the optimal decision from the perspective of energy efficiency. Since the relay node and the CH in NCS both are major energy consumers, their residual energy may not be sufficient for the subsequent monitoring. In this case, selecting them as the CH can deplete their batteries quickly, causing connec-

tion failure and subsequent clustering reconstruction. Therefore, we propose a CH selection algorithm to determine the adaptation.

1) CH SELECTION

Through the CH selection, we attempt to maximize the remaining network lifetime while maintaining network connectivity. The remaining lifetime can be defined as the interval between the current instant and the time when the first CH exhausts the energy and a cluster reconstruction is needed. Achieving this goal requires global optimization because locally selecting the node with the highest residual energy as a CH for each cluster cannot guarantee the connectivity. A possible solution may be to gather the energy of all nodes at the sink, calculate the CH selection for all clusters and disseminate the selection to all clusters. However, the information gathering and dissemination cost both time and energy; the centralized algorithm itself is inflexible to adapt to the network dynamics. Alternatively, we propose a heuristic approach to determining CHs distributively. This distributed algorithm may not obtain global optimization due to lack of global information, however, it can achieve a slightly modified goal: ensure network lifetime longer than some threshold. This threshold can be set to be the expected duration before the next switching so that the network can be alive as long as this goal can be achieved in every switching from NCS to CBS.

Guaranteeing the global connectivity by local CH selection is not an easy task. Recalling that a routing tree must be determined before CBS can be used, we perform the CH selection based on the routing tree. For a typical cluster that needs to select a CH h , a parent list is maintained as $P = \{p_1, \dots, p_k\}$, in which any node could be potentially the parent of h in the routing tree. The construction of this list will be described later in Section V-B.2. A cluster also records a child list $\mathcal{C} = \{C_1, \dots, C_l\}$, where C_j is a cluster whose CH is potentially a child of h in the routing tree. This child list is created before the switching. Each relay node randomly chooses a next-hop relay node on the routing path as a candidate parent, which in turn adds this relay node in the child list. In addition, the cluster should know all the neighboring nodes that can communicate with any node in the cluster, which can be obtained by exchanging information at the cluster creation stage. Finally, the cluster is aware of the residual energy of all cluster members, which can be easily obtained by the CH.

With the above information, we can now describe the algorithm, which is run on the relay node. The notations used in the algorithm are listed in Table 3 and the pseudo code is listed in Table 4.

The algorithm starts with the selection of the parent CH. The relay node randomly selects a candidate parent from the parent list P (line 2). Given the selected parent, the candidate CH can only be a node that can communicate with the parent. Meanwhile, to ensure the remaining network lifetime to be longer than the threshold t_l , the residual energy

TABLE 3. Notations used in the CH selection algorithm.

Notation	Definition
N	Node set of the current cluster
P	Parent list
\mathcal{C}	Child cluster list
t_l	Lifetime threshold
ϵ	Energy consumption in unit time
e_i	Residual energy in node i
B_i	Neighbor set of node i
w_i	Weight of node i

TABLE 4. CH selection algorithm.

```

1: While  $P$  is not empty do
2:    $p = \text{random\_select}(P)$ 
3:   construct set  $S = \{i | i \in N \cap B_p, e_i > \epsilon \cdot t_l\}$ 
4:   for each node  $i$  in  $S$  do
5:     for each cluster  $C_j$  in  $\mathcal{C}$  do
6:        $w_j^i = |C_j \cap B_i|$ 
7:     end for
8:      $w_i = \min w_j^i$ 
9:   end for
10:   $w = \max w_i$ 
11:   $h = \arg \max_i w_i$ 
12:  if  $w$  is not zero
13:    return  $h$  as the CH
14:  end if
15:  remove  $p$  from  $P$ 
16: end while

```

in the candidate should be more than $\epsilon \cdot t_l$, where ϵ is the energy consumption in a unit time. Thus we can obtain the candidate CH set by $S = \{i | i \in N \cap B_p, e_i > \epsilon \cdot t_l\}$ (line 3). For each node in this set, we calculate the number of nodes this node can reach in each cluster in the child list (lines 5–7) and choose the minimum number as the weight of this node (line 8). Then the node with the maximum weight is tentatively selected as the CH (line 11). The idea behind such selection is that the more nodes this CH can reach in the child clusters, the higher the probability that the child clusters can select proper CHs to ensure the connectivity. There is also a possibility that the maximum weight is zero, indicating that there is at least one child cluster in which none of the nodes can connect to a node in the candidate set. In this case, we remove the selected parent from the parent list (line 15) and repeat the above procedure until a feasible CH is selected. Otherwise, we conclude that connectivity cannot be maintained and a cluster reconstruction is necessary.

2) SWITCHING PROCEDURE

The entire switching is completed in three steps. The first step starts at the sink, which notifies all the neighboring relay nodes about the switching. For illustration purpose, we call relay nodes that take i hops to reach the sink *hop- i nodes*. These relay nodes perform the CH selection algorithm and send the selection along with the switching notification to all hop-2 relay nodes they can communicate with. The hop-2 relay nodes will then insert all the received selection in their

parent list and perform their own CH selection. This iteration will continue until all relay nodes receive the notification and complete the CH selection. Notice that if a cluster selects a CH, it also determines the parent CH in the routing tree in CBS. Then after this step, the routing tree in CBS is actually determined.

The second step starts at the leaf clusters with each relay node notifying the parent cluster the selected CH. Eventually the sink obtains the new clustering structure and perform the scheduling in CBS. Till now, the whole network is still operated in NCS and the third step performs the actual switching. Starting at the sink, each relay node notifies its children (if any) their scheduling and lets the new CH manage the cluster under CBS.

VI. EXPERIMENTAL EVALUATIONS

In this section we evaluate the performance of CBS and NCS and their integration through ns-2 simulations. For comparison purpose, we also evaluate two existing scheduling approaches used in cluster-based WSNs, for which we first give a brief description.

A. COMPARED SCHEDULING APPROACHES

The first approach we compare is a modified version of the scheduling approach used in IEEE 802.15.4, which we simply call 802.15.4 in this section. To adapt 802.15.4 in a cluster-based network, it is required to construct a cluster tree from the routing tree by adding members as the children of the corresponding CHs. Each CH or coordinator then maintains a non-conflicting superframe for its children in the cluster tree. For fair comparison, we assume the same cluster tree as in CBS are used and the length of the superframe for node i equals n_i , which is the maximum number of packets received in a cycle in CBS. The superframes are scheduled using the basic scheduling algorithm in CBS without considering the node order in the routing tree. In a superframe, we assume that each slot is collision-free so that a packet transmission in a slot never fails. When multiple children contend in a time slot, we randomly select a child to transmit while others wait for the next slot.

The second approach is a simple synchronous approach that defines a global frame for all clusters. The global frame includes intra- and inter-cluster periods. In the intra-cluster period, CHs collect data from members using the same protocol as in NCS. At the end of the intra-cluster period, all clusters enter the inter-cluster period simultaneously. The duration of intra-cluster period is also calculated in the same way as in NCS, while the duration of inter-cluster period T_o is considered as a parameter in the experiments. Although the scheduling approach is quite simple, similar ideas of this global frame have already been adopted in practice [8], [9] and we call this approach GF in the following evaluation.

B. EXPERIMENT SETUP

We first describe the cluster formation algorithm adopted before elaborating other network configurations. According to the system model in Section III-A, there are no restrictions on the cluster formation algorithm. For simplicity, in our experiments we form clusters based on their geographical positions, which are assumed already known to the nodes. Specifically, we assume the whole region is a unit square, which is divided into square cells with side length l and the sensors in the same cell form a cluster. The number of clusters is then $\lceil \frac{1}{l} \rceil^2$. The CH and relay node are randomly selected in each cluster. The transmission range is set to be $\sqrt{5}l$, which is the maximum distance between two nodes in neighbor cells. Such a range allows nodes within a cell or any two neighbor cells to communicate with each other and hence guarantees the connectivity in all clusters and the relay network.

To evaluate the network performance, we consider two networks with 300 nodes and 1200 nodes randomly scattered in the unit square. The sink is positioned at a corner of the square to create relatively long routing paths. The side length l is selected to be $\frac{1}{5}$ and $\frac{1}{10}$, resulting in 25 and 100 clusters, respectively. The cluster size ranges from 5 to 18. Some approach dependent parameters are listed in Table 5. Sensing packets have a uniform length of $30B$ and the transmission bandwidth is set to 1 Mbps. For 802.15.4 and CBS, we assume 0 header length to focus on comparison of cycle scheduling. For NCS and GF, we adopted the default header length of MAC 802.11 in ns2, which are 44 B, 38 B, 52 B and 44 B for RTS, CTS, DATA and ACK. The performance metrics evaluated are packet delay and network throughput. Packet delay is defined as the average end-to-end delay for all packets received at the sink while the network throughput can be interpreted as the maximum packet generation rate with which the network can operate steadily. The evaluation time is set to 100 seconds to obtain the network performance at the stable state. Each experiment is repeated 10 times to obtain the average value.

TABLE 5. Parameters of approaches.

Approach	Parameter	300-node network	1200-node network
NCS	T_c	0.2s	2s
GF	T_o	2s	16s

The inter-cluster communication in NCS and GF utilizes the common IEEE 802.11 MAC protocol. Practical WSNs may adopt some simplified versions of 802.11, however, the variation among these versions only affects the inter-cluster communications but does not substantially affect the overall performance evaluation. We thus adopt the default controlling packet length of MAC 802.11 in ns2, which is 44 B, 38 B, 52 B and 44 B for RTS, CTS, DATA header and ACK, respectively. For TDMA based 802.15.4 and CBS, we assume 0 header length to focus on comparison of cycle scheduling. To

construct the routing tree, the CH or the relay node randomly selects a node in the neighbor cells that are closer to the sink as its next-hop. The aggregated packets at the source are not further aggregated at the intermediate nodes in the routing tree.

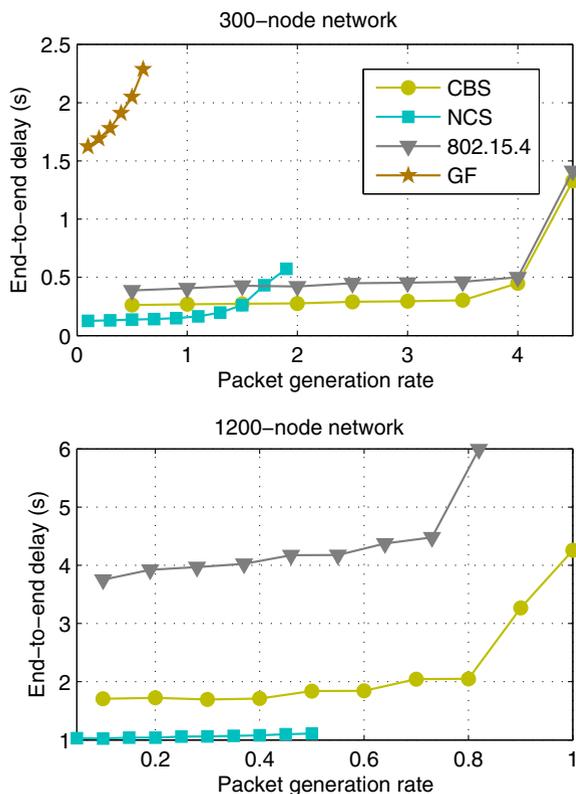


FIGURE 4. End-to-end packet delay of four scheduling approaches under different packet generation rates.

C. PERFORMANCE RESULTS

Fig. 4 shows the average packet delay among four approaches with allowable packet generation rates. The standard deviations of these delays are less than 0.02 and 0.2 for 300-node and 1000-node network respectively, indicating stable performance of the examined approaches. We first examine the result in the network with 300 nodes. We can observe that NCS yields the shortest packet delay when the network is not saturated under lower packet generation rates. Due to the introduction of relay nodes, packets are transmitted quickly at each hop without undertaking any extra delay caused by the state switching. On the opposite, GF, which also uses 802.11 for inter-cluster communications, yields the longest packet delay. In GF, the synchronization of the inter-cluster periods for different CHs incurs many concurrent packet transmissions with high contentions, eventually resulting in long delay. Two TDMA based approaches have shorter delay than GF since they completely avoid the transmission contention. However, since the transmissions in these two approaches are strictly scheduled, packets inevitably incur some queuing

delay before they can be relayed by the intermediate nodes in the routing path. Thus both have longer delay than NCS. In particular, delay in CBS is about 30% shorter than that in 802.15.4, due to the efficient design of the cycle scheduling. While the cycle length of two approaches does not have much difference, packets in CBS endure less queuing delay with the ordered interval scheduling.

The performance comparison is similar in the network with 1200 nodes, where GF exhibits poor performance with allowable generation rate under 0.05, which was not shown in the figure. While NCS still shows the shortest delay, we observe that CBS obtains a higher performance gain compared to 802.15.4, whose delay is nearly twice of CBS. This contrast indicates that the scheduling in CBS enjoys more benefits in larger-scale networks.

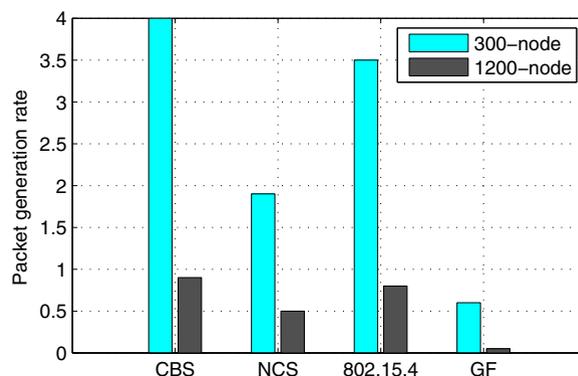


FIGURE 5. Network throughput with four scheduling approaches.

Fig. 5 shows the network throughput, or the maximum packet generation rate for four approaches in both networks. GF has the worst throughput, besides its longest delay as seen in Fig. 4. This is because that GF requires a long inter-cluster period to accommodate long delay and therefore causes low allowable packet generation rate. NCS also has lower throughput compared to two TDMA-based approaches. This is mainly due to the intrinsic of CSMA-based 802.11 protocol, which spends much longer time than the actual transmission time in transmitting packets. For two TDMA-based approaches, CBS slightly outperforms 802.15.4. The similar performance is due to the fact that the cycle length, a dominating factor for the throughput, is similar in both approaches.

Recall in NCS, the data collection duration affects the number and length of aggregated packets and eventually the maximum achievable throughput. In the next experiment, we further reveal the relationship between the maximum generation rate and the data collection duration in Fig. 6. Clearly, the maximum generation rate increases faster when the duration is relatively short. Specifically, it increases about 2 times when the duration is increased from 0.1 s to 0.5 s, and only 14% when the duration continues to increase to 1 s. Since increasing the data collection duration directly increases the collection delay, such observation indicates that the data

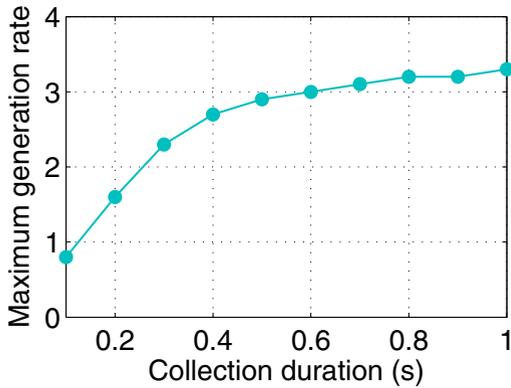


FIGURE 6. Maximum generation rate achieved with different collection durations in NCS.

collection duration should be chosen properly to achieve the best tradeoff between the maximum generation rate and the packet delay.

D. INTEGRATION PERFORMANCE

We evaluate the integration performance in the network with 300 nodes. In the integration, we are concerned with the achievable network lifetime, which is defined as the duration between the network creation and the time instant when the relay network is no longer connected. This terminating situation corresponds to the following three cases: 1) the first node depletes its energy during data gathering; 2) a cluster cannot complete the CH selection algorithm in Section V during the switching from NCS to CBS; and 3) A relay node or a CH loses connection with its parent during the switching.

To evaluate the network lifetime, we simulate the switching by setting the consecutive time durations in CBS and NCS to be randomly distributed between 30 ~ 90 minutes and 30 ~ 90 seconds, respectively. The packet generation rate is fixed at 1 packet/s. To model the energy consumption, we assume the power ratio for sending, receiving and idle listening is set to 1.67:1:0.88 as adopted in [14]. In CBS, we assume each data packet is associated with an ACK packet. The length of all control packets is of default values in ns-2. The initial energy of every node is normalized to energy consumed in 24-hour continuous packet receiving. This value is reasonable if we assume the receiving power of a sensor is of order of 10mW and the battery energy is of order of 1000J. Similarly, the threshold energy is set to the energy consumed in 90-minute packet receiving.

With the above parameters, network lifetime is heavily affected by the performance of the CH selection algorithm, which in turn is affected by the transmission range. A too small transmission range will cause failure of network connection and hence the CH selection algorithm. On the other hand, when the transmission range is set to $\sqrt{5}l$, the connectivity is guaranteed in the experiment and the CH

selection algorithm can be reduced to selecting the node with the highest residual energy. Thus we evaluate the selection algorithm by varying the transmission range to change the connectivity. Here we assume the transmission power does not change with different transmission ranges to focus on the performance of the CH selection algorithm. In addition, since network lifetime could be as long as several hundreds of hours, we simplify the simulation such that in each CBS duration, we only simulate the network for 1 minute and use the obtained energy consumption to estimate the actual energy consumption for the whole duration. That is, if the CBS duration lasts for m minutes and a node consumes energy e in a minute, its total energy consumption for this duration is $m \cdot e$. We expect such approximation will not cause much difference on the results because of the fixed packet generation rate and hence the stable traffic pattern.

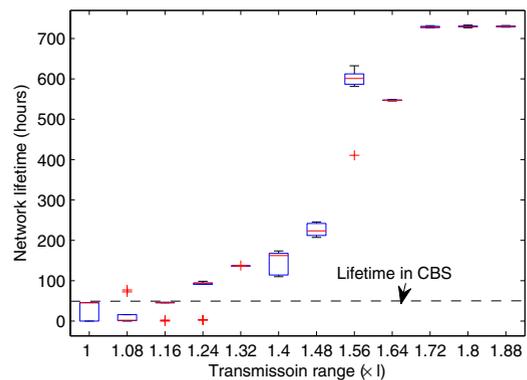


FIGURE 7. Box plot of network lifetime under different transmission ranges. The dashed line represents the network lifetime when CBS is adopted.

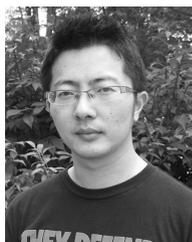
Fig. 7 shows the network lifetime under different transmission ranges. We use the box plot to better reveal its variation. For comparison, the dashed line represents the lifetime when only CBS is adopted and CHs cannot be changed. It can be seen that when the transmission range is short and the selection of CH to maintain connectivity is limited, the lifetime with scheduling switching is similar to or sometimes worse than the lifetime under CBS. In this case, transmissions in NCS consume more energy and lower the lifetime. However, when the transmission range slightly increases, its benefit becomes more evident with lifetime growing exponentially. When the range is above $1.72 \cdot l$, the lifetime reaches a maximum at about 720 hours, approximately 16 times of the lifetime under CBS. Notice that 16 is the number of nodes in the cluster closest to the sink, which consumes more energy than other clusters. This maximum lifetime demonstrates that the algorithm can efficiently and fairly dissipate the energy consumption among all possible nodes in the cluster to maximize network lifetime. On the other hand, the increasing network lifetime also indicates that the switching between CBS and NCS does not affect the connectivity of the network.

VII. CONCLUSION

In this paper, we have presented a hybrid scheme that integrates two communication scheduling approaches CBS and NCS to enable cluster-based WSNs to serve as network infrastructure of information collection in CPS. In CBS, a cycle based schedule for each CH is constructed based on the pre-determined routing tree. CBS minimizes the cycle length while maintaining the node order in the routing tree, which minimizes the intra-cluster collection delay and allows continuous packet forwarding from the source to the sink. In NCS, a CH-relay-member structure is proposed to replace the conventional CH-member structure. The introduction of relay nodes releases the CHs from the heavy burden of packet relaying so that the intra- and inter-cluster communications can be performed more efficiently. Our simulation results have shown that the proposed approaches exhibit much better performance than existing scheduling approaches in terms of packet delay and throughput. The hybrid scheme integrates CBS and NCS without any interruption on data gathering during switching, allows the network to enjoy the benefits of both approaches to meet the stringent requirement for CPS.

REFERENCES

- [1] L. Sha, S. Gopalakrishnan, X. Liu, and Q. Wang, "Cyber-physical systems: A new frontier," in *Machine Learning in Cyber Trust*. New York, NY, USA: Springer-Verlag, 2009, pp. 3–13.
- [2] G. Gupta and M. Younis, "Load-balanced clustering in wireless sensor networks," in *Proc. IEEE ICC*, May 2003, pp. 1848–1852.
- [3] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Annu. Hawaii Int. Conf. Syst. Sci.*, Jan. 2000, pp. 1–10.
- [4] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 366–379, Oct./Dec. 2004.
- [5] Z. Zhang, M. Ma, and Y. Yang, "Energy efficient multi-hop polling in clusters of two-layered heterogeneous sensor networks," *IEEE Trans. Comput.*, vol. 57, no. 2, pp. 231–245, Feb. 2008.
- [6] D. Gong, Y. Yang, and Z. Pan, "Energy-efficient clustering in lossy wireless sensor networks," *J. Parallel Distrib. Comput.*, vol. 73, no. 9, pp. 1323–1336, 2013.
- [7] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Comput. Commun.*, vol. 30, nos. 14–15, pp. 2826–2841, 2007.
- [8] M. Lotfinezhad, B. Liang, and E. S. Sousa, "Adaptive cluster-based data collection in sensor networks with direct sink access," *IEEE Trans. Mobile Comput.*, vol. 7, no. 7, pp. 884–897, Jul. 2008.
- [9] H. J. Choe, P. Ghosh, and S. K. Das, "QoS-aware data reporting control in cluster-based wireless sensor networks," *Comput. Commun.*, vol. 33, no. 11, pp. 1244–1254, 2010.
- [10] L. Shi and A. O. Fapojuwo, "TDMA scheduling with optimized energy efficiency and minimum delay in clustered wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 7, pp. 927–940, Jul. 2010.
- [11] F. Cuomo, E. Cipollone, and A. Abbagnale, "Performance analysis of IEEE 802.15.4 wireless sensor networks: An insight into the topology formation process," *Comput. Netw.*, vol. 53, no. 18, pp. 3057–3075, 2009.
- [12] 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.
- [13] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proc. ACM SenSys*, 2003, pp. 171–180.
- [14] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for tree-based data gathering in sensor networks," *Wireless Commun. Mobile Comput.*, vol. 7, no. 7, pp. 863–875, 2007.
- [15] S. Liu, K. Fan, and P. Sinha, "CMAC: An energy efficient MAC layer protocol using convergent packet forwarding for wireless sensor networks," in *Proc. IEEE SECON*, Jun. 2007, pp. 11–20.
- [16] J. Kim, X. Lin, N. B. Shroff, and P. Sinha, "On maximizing the lifetime of delay-sensitive wireless sensor networks with anycast," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 807–815.
- [17] J. Kim, X. Lin, and N. B. Shroff, "Optimal anycast technique for delay-sensitive energy-constrained asynchronous sensor networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 2, pp. 484–497, Apr. 2011.
- [18] 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.
- [19] P. A. W. Lewis and G. S. Shedler, "Simulation of nonhomogenous Poisson processes with log-linear rate function," *Biometrika*, vol. 63, no. 3, pp. 501–505, 1976.
- [20] S. C. Ergen and V. Pravin, "TDMA scheduling algorithms for wireless sensor networks," *Wireless Netw.*, vol. 16, no. 4, pp. 985–997, 2010.
- [21] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: A stateless protocol for real-time communication in sensor networks," in *Proc. ICDCS*, May 2003, pp. 46–55.
- [22] E. Felemban, C. G. Lee, and E. Ekici, "MMSPEED: Multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 6, pp. 738–754, Jun. 2006.
- [23] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichertiu, "Z-MAC: A hybrid MAC for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 511–524, Jun. 2008.
- [24] G. S. Ahn, S. G. Hong, E. Miluzzo, A. T. Campbell, and F. Cuomo, "Funnelling-MAC: A localized, sink-oriented MAC for boosting fidelity in sensor networks," in *Proc. 4th ACM Conf. Sensys*, 2006.



XI DENG received the M.S. degree in computer science and engineering from Tsinghua University, Beijing, China. He is currently pursuing the Ph.D. degree in computer engineering with Stony Brook University, Stony Brook, NY, USA. His current research interests include network simulation methodology and delay sensitive wireless sensor networks.



YUANYUAN YANG received the B.Eng. and M.S. degrees in computer science and engineering from Tsinghua University, Beijing, China, and the M.S.E. and Ph.D. degrees in computer science from Johns Hopkins University, Baltimore, MD, USA. She is a Professor of computer engineering and computer science with Stony Brook University, Stony Brook, NY, USA, and the Director of Communications and Devices Division, New York State Center of Excellence in Wireless and Information Technology. Her current research interests include wireless networks, data center networks, optical networks, and high-speed networks. She has published more than 260 papers in major journals and refereed conference proceedings and holds seven U.S. patents. She is currently an Associate Editor-in-Chief for the IEEE TRANSACTIONS ON COMPUTERS and an Associate Editor for the *Journal of Parallel and Distributed Computing*. She has served as an Associate Editor for the IEEE TRANSACTIONS ON COMPUTERS and the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. She has served as a General Chair, Program Chair, or Vice Chair for several major conferences and a Program Committee Member for numerous conferences.