Monitoring power transmission lines using a wireless sensor network
Junru Lin1,2, Baohui Zhu1,2, Peng Zeng1, Wei Liang1, Haibin Yu1* and Yang Xiao3

1 Key Lab. of Networked Control Systems, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016, China
2 University of Chinese Academy of Sciences, Beijing, 100049, China
3 Department of Computer Science, The University of Alabama, 101 Houser Hall, Box 870290, Tuscaloosa, AL 35487-0290, U.S.A.

ABSTRACT
Power transmission is the bulk transfer of electrical energy from power plants to sub-stations. A wireless sensor network is a promising technology for transmission line monitoring due to its low cost, easy installation, large-scale coverage, and fault tolerance characteristics. A wireless sensor network is application-specific; therefore, we investigate the new features and requirements of the wireless sensor network used in transmission line monitoring. Then, we propose an efficient wireless sensor network framework, which includes a clustering algorithm to simplify network management and to balance the network’s energy consumption and a hybrid media access control (MAC) (H-MAC) protocol to handle traffic variability. The framework takes advantage of the features of network topology and traffic pattern to optimize the protocols’ performance on real time and energy efficiency. The results indicate that the H-MAC shows a significant improvement in the network’s reliability, real-time performance, and energy efficiency, and the cluster hierarchy can balance the network’s energy consumption. Furthermore, the cluster hierarchy also prolongs the network’s lifetime. Copyright © 2014 John Wiley & Sons, Ltd.

KEYWORDS
transmission line monitoring; smart grid; wireless sensor networks

*Correspondence
Haibin Yu, Shenyang Institute of Automation, Chinese Academy of Sciences, No.114 Nanta Street, Shenehe District, Shenyang, Liaoning Province, 110016, China.
E-mail: yhb@sia.cn

1. INTRODUCTION
Power transmission is the bulk transfer of electrical energy from power plants to sub-stations, which are a critical part of power grid. Security and reliability of power transmission have significant effects on the stability of the whole grid. Transmission line online monitoring is a powerful tool for transmission line protection and diagnosis [1–3], and there are many factors that can threaten the security of the transmission grid infrastructure such as extreme weather conditions, the surrounding environment, artificial destruction, and overheating. Based upon the transmission line monitoring, we can detect and respond to threats before services are impacted. We can also find the fault location rapidly, enable state-based maintenance, and repair the fault. Transmission line online monitoring is also a powerful tool for maximizing the capabilities of the transmission line. The load capability of a transmission line is determined when the lines are designed. To guarantee the reliability of the grid, the transmission system needs capability of handling an emergency. Based upon the parameters of the transmission line monitoring, a load can be forecasted [4] and the real-time dynamic current capacity of overhead power lines can be evaluated [5,6]. Typically, the dynamic rating provides a higher line capability of 98% of the time, and it also provides at least 15–30% additional capability for over 95% of the time [7]. Therefore, we can utilize the power lines more effectively by planning and operating power transmission in a dynamic way instead of being under static, conservative assumptions.

Nowadays, the transmission line becomes longer and more dispersed; some segments may travel through areas with complicated terrain conditions. As a result, transmission line maintenance becomes increasingly difficult, and most transmission line monitoring and supervision are carried out by humans, mobile robotic, or helicopters. All these methods have a high cost and high labor intensity. Additionally, transmission line conditions cannot be monitored continually, and all of the transmission lines cannot
be covered at one time. Using digital cellular technology, such as Global System For Mobile Communication (GSM), Worldwide Interoperability for Microwave Access (WiMAX), and Code Division Multiple Access (CDMA), to develop a transmission line online monitoring system becomes popular because it provides a way to monitor transmission lines at any time. Although the cost of using only digital cellular technology to monitor transmission lines is much lower than the aforementioned conventional methods, it is still too expensive to implement a grid-wide monitoring system [8]. Only a fraction of the whole transmission infrastructure is considered to be critical and will be covered by the online monitoring system.

Characteristics, such as low cost, easy installation, large-scale coverage, and fault tolerance, make the wireless sensor network a promising part of technology for transmission line online monitoring [8–10]. The scenario of transmission line monitoring brings the wireless sensor network new features and requirements. First, the deployment of the transmission grid infrastructure makes the wireless sensor network have a linear topology. Second, most sensors are placed around the supports, and as a result, the network becomes dense at that place, whereas it is sparse in most other areas. Third, traffic exhibits a large amount of variability. When the network collects periodic sensing information, traffic becomes much more congested because all of the sensors attempt to transmit their sensing data. At other times, the traffic is very light, and most of it is generated by alarm, network control, and configuration. Finally, the traffic load on different nodes may be unbalanced because of the nodes’ sensing functions or roles in the network. By now, most of the work on wireless sensor networks that was developed for transmission line monitoring [2,11] uses Zigbee/IEEE 802.15.4 as a means of communication technology. These works have not considered the features’ impact on network performance, and it also has not utilized them to optimize the network performance. We proposed an efficient wireless sensor network framework for transmission line online monitoring, and this paper makes the following three contributions:

- On the basis of the network topology characteristics, we propose a cluster hierarchy to simplify network management and a dense-set-based cluster rotation algorithm to resolve the problem of intra-cluster energy unbalance. A location-aware address assignment scheme and an address-based routing protocol are developed to support cluster hierarchy.
- We design a hybrid MAC protocol, H-MAC, to handle traffic variability. In H-MAC, we use S-XMAC, which takes advantage of loose synchronization to reduce the number of preambles sent before data packet transmission, and we also propose a location-based wake up schedule to optimize the real-time performance of uplink data. In the busy period, we use pipelined MAC to deal with the hidden terminal problem and to support the sensors with large volume of data.
- We evaluate the H-MAC and clustering algorithm by simulations. Simulation results show that H-MAC can adapt to variable traffic, meet the real-time and reliability requirements of the application, and make the network energy efficient at the same time. Also, the clustering algorithm resolves the intra-cluster unbalance problem, and it significantly prolongs the network’s lifetime.

The rest of the paper is organized as follows. In Section 2, we review the prior efforts of using wireless sensor networks in transmission line monitoring. In Section 3, we discuss issues related to the architecture of wireless sensor networks, and then, we analyze the requirements and features of the application. We present our wireless sensor networks framework for transmission line monitoring. In Section 4, we present a dense-set-based cluster hierarchy. In Section 5, we present a location-aware addressing scheme and an address-based routing protocol. In Section 6, we present H-MAC. The performance of the wireless sensor network framework under various test environments is presented in Section 7. Finally, we conclude this paper in Section 8 and describe on-going directions.

2. RELATED WORK

Smart grid provides efficient, reliable, automated, and intelligent electrical services based on information of the whole grid, such as power generation, transmission, distribution, and consumption. Therefore, two-way, cyber-secure communication technologies play a vital role in smart grid [3,12–20]. Significant work on information technologies in smart grid has been carried out both in academia and industry. The majority of this work only emphasized the distribution grid and the demand side leaving the big picture of transmission grid in the context of smart grids unclear [21]. In smart transmission grid online monitoring, the transmission grid spreads over a large area and has a huge number of assets. In the meantime, wired methods require expensive cables, and this makes it impracticable to monitor every asset grid widely. Besides the dedicated wired communication and power line communication, wireless communication can also be potential for transmission line monitoring. Power line communication technology uses power transmission wires as its media, and it is mainly applied in indoor environment [22]. As for transmission line monitoring, broadband over power line technology can be a potential technology. If any of the support fails or if the transmission line breaks, broadband over power line communication fails as well, and this makes it difficult to find fault location. Nowadays, wireless technology has already been used in transmission line online monitoring. Cellular technologies, such as GSM, CDMA, and WiMAX, are the most widely used [23,24], and most of the systems described in the literature have a limited number of monitoring points due to their cost. Only a fraction of the whole transmission infrastructure that is considered to be “critical span” will be monitored [9].
The wireless sensor network becomes a promising technology for transmission line online monitoring due to its low cost, easy installation, large-scale coverage, and fault tolerance characteristics [8–10]. There have been some experimental studies on performance of wireless sensor network platforms in the power system environment [10,25–52], and the paper [25] analyzes the performance of the Mica2 motes in a linear aligned wireless sensor network for long distance overhead power lines. The paper [10] presents an experimental study on the wireless link assessment in different electric power systems including a 500 kV sub-station. Field test results provide valuable insight about 802.15.4 compliant platforms for wireless sensor networks in smart grid applications. Both experiments show that Radio frequency (RF) technology of wireless sensor network platforms has potential to be used in power transmission monitoring system. Young-II Kim et al. [53] implemented a prototype system for facility management by using wireless sensor networks. In this prototype, sensor nodes are installed on four 765 kV power towers to monitor transmission towers. Every power tower has a sink node that collects sensing data from ZigBee modules and that relays the data to the sub-station by wireless LANs. This method can monitor assets around the sub-stations, but it cannot be used to monitor all assets through power lines hundreds of thousands of miles long. Yang et al. [9,11] proposed a distributed power line sensor-net to discover cost-effective monitoring. In the sensornet, sensing data can be delivered to a sub-station hop by hop, and this causes the sensornet to have a large coverage. They also used the sensornet to monitor the asset’s status, and then, they evaluated the real-time dynamic current capacity of overhead lines to maximize the utilization of existing power grid [5]. Hybrid architectures, which use both wireless sensor technology and other wireless technologies, such as cellular networks, are considered to be more efficient, more robust, and real time [2,3]. Leon et al. [1] propose a two-layer model of wireless sensor networks for secure electric energy infrastructure to overcome the restrictions imposed by the range/energy management issue on sensor nodes. The model uses nodes with a communication range less than 100 feet to form a local group on each support. Each support has an enhanced node with a larger communication range. Enhanced nodes form the second layer, which handles and delivers all messages to the sub-station. Aravinthan et al. [54] propose a three-layer architecture to increase reliability and reduce the latency of event notifications for the distribution feeder, which introduces an additional layer based upon the two layers in Leon’s model. The third layer adopts WiFi to communicate with sub-stations to increase the redundancy and reliability. Huang et al. [2] propose a reconfigurable wireless sensor network model for overhead transmission line monitoring where sensor/relay nodes can also communicate with other nodes using the cellular network, and relay nodes can collaborate to turn on or off GSM/GPRS/UMTS devices depending on the application requirements and configurations.

The majority of the work with using the wireless sensor network for transmission line monitoring did not investigate how the network forwards data or just assumed that an underlying network, for example, a ZigBee network. Additionally, performances of wireless sensor network were not evaluated in most of the previous works. Because the wireless sensor network is application-specific in nature, it is necessary to develop an optimized wireless sensor network protocol, which takes specific characteristics into account.

### 3. ARCHITECTURE

The transmission line online monitoring system includes conductor temperature measurement, dynamic thermal capacity forecasting, mechanical strength of support measurement, conductor sag measurement, environmental micro-meteorological monitoring, and so on. Sensors commonly used are listed in Table I. Wind speed sensors, temperature sensors, wind direction sensors, humidity sensors, and rain sensors are mainly utilized to measure weather parameters. On the basis of these parameters, we can see how electricity demand fluctuates as weather changes and forecast electricity load [4]. Because weather conditions change slightly in a certain area, only part of the supports need these types of sensors, and all of these sensors are attached to supports (position A). The sensors to measure

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Total size in data collection per monitoring cycle</th>
<th>Installation position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wind direction sensor</td>
<td>4</td>
<td>A</td>
</tr>
<tr>
<td>Temperature sensor</td>
<td>4</td>
<td>A</td>
</tr>
<tr>
<td>Wind speed sensor</td>
<td>4</td>
<td>A</td>
</tr>
<tr>
<td>Humidity sensor</td>
<td>4</td>
<td>A</td>
</tr>
<tr>
<td>Rain sensor</td>
<td>4</td>
<td>A</td>
</tr>
<tr>
<td>Strain sensor</td>
<td>8</td>
<td>A/B</td>
</tr>
<tr>
<td>Accelerometer for tilting</td>
<td>8</td>
<td>A/B</td>
</tr>
<tr>
<td>Conductor temperate sensor</td>
<td>4</td>
<td>B</td>
</tr>
<tr>
<td>Accelerometer for vibration</td>
<td>4</td>
<td>A/B</td>
</tr>
<tr>
<td>Magnetic field sensor for current</td>
<td>4</td>
<td>B</td>
</tr>
<tr>
<td>Accelerometer for line galloping monitoring</td>
<td>5120</td>
<td>C</td>
</tr>
<tr>
<td>Magnetic field sensor for power quality graph</td>
<td>4000</td>
<td>B</td>
</tr>
</tbody>
</table>

Table I. Typical sensors used in transmission line monitoring.
the conditions of supports and other assets such as vibration sensors and tilt sensors should be installed on every asset. Otherwise, it is difficult to find fault location exactly. These types of sensors are also attached to supports/poles. Sensors for conductor monitoring, such as the temperature sensor, strain sensor, tilt sensor, and vibration sensor are placed on both end points of each transmission line at every supports (position B). There are also some types of sensors that are placed on transmission lines (position C), such as accelerometer sensors for transmission line galloping monitoring. Although most sensors are attached to supports or somewhere very close to the supports, the overview of the sensor’s placement is shown in Figure 1.

In the transmission line monitoring application, the network and traffic have some application-specific features, which are the keys to improving the network performance to meet application requirements. The network model and traffic model of our wireless sensor network framework are as follows.

### 3.1. Network model

There are a number of supports between sub-stations. The distance between two adjacent supports ranges from 400 to 800 m depending on the geographical constraints, whereas the distance between adjacent sub-stations can be 50 km. Therefore, there are 60 ~ 125 supports between two sub-stations. Supports are roughly aligned as a line, and as a result, the network has a linear topology.

Because transmission lines are long and dispersed, the number of assets that need to be monitored is large. In the hybrid architecture, using cellular technology, such as GSM/General Packet Radio Service (GPRS)/Universal Mobile Telecommunications System (UMTS)/WIMAX/CDMA, to assist the wireless sensor network in forwarding data, is a more efficient and robust solution in transmission line monitoring compared with using the wireless sensor network alone. This is so because, first, wireless sensor network is a low data rate network; however, the transmission line monitoring system has many sensor nodes including those with large volume of data. It is hard to use the wireless sensor network alone to forward data efficiently. By using cellular modules, nodes can communicate with control center directly. As a result, not only does the parallelism of the network increase but also the traffic of wireless sensor network is reduced, especially when sensors with large volume of data are placed near cellular modules. Second, cellular technology can help balance the network’s traffic and energy consumption. If one is using the wireless sensor network only, nodes closer to sub-stations have to forward more traffic and consume more energy. Cellular modules can help alleviate the imbalance because the network can route data through cellular modules. Finally, utilizing cellular technology can improve network performance for being real time and reliable. In a linear network traveling through areas with complex terrain conditions, nodes fail easily because of the potential security attack, and then, they cause parts of the network to disconnect with the control center. Cellular technology can isolate the failure and improve the network’s real-time performance, especially the nodes in the middle of the network because they need less hops to reach the control center by routing through the node with cellular modules.

In our framework, we adopt a two-layer network model shown in Figure 2; three types of nodes are defined:

- **Basic sensor node (BSN):** They are the most common nodes in the network. They sense the transmission line conditions and communicate with other nodes. They are equipped with a middle range wireless transceiver.
- **Data relay node (DRN):** They are basic nodes without a sensing function. They are only used for relaying packets. They help improve the network’s reliability and prolong the network’s lifetime.
- **Enhanced relay node (ERN):** They collect sensing data and deliver them to the control center directly. Each ERN is equipped with not only a transceiver with middle range but also a cellular module. They use a wireless communication radio to communicate with BSNs and DRNs to collect sensing data and then send the sensing data to the control center via the cellular communication radio.
The architecture consists of two layers. The BSNs, DRNs, and ERNs form layer 1. In layer 1, every node communicates with each other using the middle range transceiver whose communication range is between 400 and 800 m. ERNs act as sink nodes in this layer, and BSNs deliver their sensing data to the nearest ERN in a hop by hop manner. Layer 2 is composed of ERNs and the control center. The control center is the sink node of the whole network. It collects, processes, and analyzes all the sensing data from the transmission line monitoring system and then makes decisions and suggestions intelligently. The ERNs forward the data collected in layer 1 to the control center with their cellular communication modules. Our two-layered architecture is different from the multi-layer architecture proposed in [1,54]. First, we use cellular technology to communicate with the control center directly in layer 2 rather than using the middle range radio to connect with the control center in a hop by hop manner. In this way, we can take advantage of the hybrid architecture’s advantage, which is mentioned previously. Second, in layer 1, the restrictions imposed by the communication range on the sensor node are relaxed. We consider that the radio has a communication range between 400 and 800 m rather than 100 feet. There are two reasons why we can choose such a radio in transmission line monitoring. On one hand, in transmission line monitoring, sensor nodes are deployed outdoors, and in this line-of-site environment, the wireless transceiver can have a longer communication range. It is only a little extra hardware cost in a wireless module with a middle communication range compared with a wireless module with a short range. There are also some commercial products [55] whose communication range can meet the requirements of our architecture already. On the other hand, when every node is equipped with a middle range wireless module, node failure can be easily fixed by using other nodes to forward the data, and the network becomes more reliable and robust in this way.

The most significant characteristic of layer 1 is its linear chain topology, but there are still some differences from the linear network studied in literatures where nodes are deployed in a random manner [56] or a simple linear manner [57]. The differences brought by the application are as follows:

- **local high density**: most sensors utilized in transmission line monitoring are attached to supports. Just a small part of the sensors are deployed in transmission lines between the supports. Therefore, the network is dense around the supports, whereas sparse in other locations.
- **stability**: there is no mobile node in the network and no movement caused by wind, sagging, and galloping that can be ignored because the shifting distance is too short compared with the radio’s communication range. Therefore, we can consider the network to be stable once it has been deployed.

### 3.2. Traffic model

There are two types of data in transmission line monitoring, one is periodic data, which is generated by sensors and collected every 15 to 30 min. The other is aperiodic data, which appears rarely in normal and healthy situations. This type of data is mainly generated by alarm, network control, or user-specified operations. As a result, the system’s duty cycle (15 to 30 min) can be divided into two parts, the busy period and idle period. The busy period starts by ERN starting periodic data gathering and ends by every sensor node’s sensing data arriving at the ERN; this period lasts a short time, but the traffic is heavy because every sensor node attempts to deliver its sensing data. The idle period is the rest of duty cycle excluding the busy period. Compared with the busy period, this period lasts much longer, but traffic in this period is much lighter.

It should be noted that traffic generated by sensors with large volume of data is not included in the traffic model because this type of sensors is used for critical span monitoring only. Also, in a critical span, there is always an ERN, which has sufficient energy supply and a more powerful computing capability. Sensors can send data to the ERN directly when they are polled, and this will not affect layer 1 much if polled in a proper way. Therefore, we overlook that traffic in our work; however, the evaluation shows that the network also goes well if there is a small number of that type of sensor in the network.
3.3. System requirements

In transmission line monitoring system, only ERNs have solar cell batteries, whereas other nodes generally use a high-energy lithium battery due to the requirement of low cost and miniaturization. However, each node is expected to run 5 years at least. Therefore, the network needs to be energy efficient.

Timing is critical in smart grid communications, and messages between different entities within smart grid have different latency requirements. The requirement of network latency in transmission line monitoring is much lesser than in other parts of the smart grid such as sub-station automation. Note that the supervisory control and data acquisition system cycle is typically 4 s [1], and the delay of the urgent message can be two times larger than that of the supervisory control and data acquisition system cycle period [2]. In the IEEE 1646 standard, the communication timing requirement of monitoring and control information external to the sub-station is 1 s, whereas operations and maintenance information external to sub-station is 10 s. Urgent messages about unusual events have more stringent real-time requirements. These timing requirements are still challenges for the wireless sensor network, which has limited energy and multiple hops in this application.

To adapt to the features of the network model and traffic model, we propose an efficient wireless sensor network framework to meet the system requirements. The framework is based upon cluster hierarchy, which is integrated with a location-aware address assignment scheme, an address-based routing protocol and an H-MAC protocol. The goal of the framework is to provide energy efficient communication under variable traffic while meeting real-time requirements for different types of data by taking advantage of the characteristics of the network model.

4. CLUSTER HIERARCHY

Because of the constraint of the transmission grid infrastructure, the topology of the wireless sensor network is linear, and most of the sensors are attached to supports/towers. With cluster hierarchy, sensor data from the same clusters can be aggregated, and this reduces the number of channel access, and it saves energy. Cluster heads can be used to guarantee the real-time performance of the network, whereas other nodes stay in low power consumption modes. Furthermore, compared with the flatted-network routing, the cluster-based routing does not need complicated route maintenance, which gives the network great scalability.

With cluster hierarchy, the network is energy unbalanced in two aspects: (i) intra-cluster energy unbalance; compared with the cluster members, the cluster heads have to undertake more tasks, such as data processing, packets forwarding, and route maintenance, and this way results in more energy consumption; (ii) inter-cluster energy unbalance; the “many to one” type of communication patterns cause a funnel effect especially in a linear network, and the cluster closer to the sink needs more energy to forward more packets. This makes it become the bottleneck of the network’s lifetime.

We propose a clustering algorithm to solve the intra-cluster energy unbalance, and we also take advantage of the linear network’s potential position information to design a location-aware node naming scheme and routing protocols. Both the clustering algorithm and routing protocols are scalable. Therefore, we can alleviate the funnel effect just by deploying relay nodes.

There have been some clustering algorithms developed for the wireless sensor network. LEACH [58] selects cluster heads randomly, but it does not take energy consumption of each node into account. Cluster heads need to communicate with the sink directly rather than in a hop by hop way, and this causes the networks to have limited coverage. Also, nodes further away from the sink consume more energy. LEACH_C and LEACH_F [58] improve LEACH by finding a more suitable cluster head with a centralized scheme, but this scheme needs to collect the network’s global information in each round, and this is a large overhead for the wireless sensor network. PEGASIS [59] further utilizes the greedy algorithm for cluster head selection and reduces the energy consumption based on chain topology. HEED [60] extends LEACH by incorporating residual energy and intra-cluster communication cost to determine each node’s probability of becoming a cluster head. In the application of transmission line monitoring, the network topology is more regular than the scenario considered in LEACH, LEACH_C, LEACH_F, and HEED. By taking advantage of linear and local-high-density topology, we propose a dense-set-based clustering algorithm. Incorporating the location-aware naming scheme and routing protocol, little topology management and route maintenance are needed after cluster head rotation. To achieve this effect, the MAC protocol needs to provide some appropriate supports, which have been implemented in our MAC protocol.

The main procedures of dense-set-based clustering algorithm include three phases, which are stated in the following sections.

4.1. Dense set construction phase

The dense set construction process occurs when nodes join the network. This phase’s main mission is making sure that all nodes attached to the same support/tower form a set called the “dense set”. In the dense-set-based clustering algorithm, in order to form a cluster, a node is chosen from the current dense set as the cluster head, and all nodes of the next dense set act as cluster members. The dense set is the basis of other phases of the clustering algorithm, especially in the cluster head rotation phase. The dense set construction procedure includes the following:

1. When a node starts to join the network, it obtains information about dense sets around it by broadcasting a request to the dense set.
If the joining node receives responses, it decides which dense set to join in by the received signal strength indicator (RSSI) value of responses. It can be easy to distinguish whether the packets are from nodes attached to the same support or not by using the threshold scheme because the distance from the sensor nodes in different supports is much longer than the distance from nodes in the same support. This makes the RSSI value of packets from nodes in the same support much greater. If a joining node receives a response from a node in the same support/tower, it joins the dense set indicated by the response. Otherwise, it establishes a new dense set and becomes a cluster head of the new dense set. When joining a dense set, a node broadcasts its information to that dense set and every member of that dense set, then adds it to their dense set member lists.

After this phase, all the sensor nodes attached to the same supports form a dense set. Figure 3 shows a graphic representation of the clusters and dense sets generated in this phase.

### 4.2. Data collection

Data collecting includes periodic data collection in the busy period and aperiodic data collection in the idle period. With cluster hierarchy, a cluster head belongs to two different clusters at most. It is the cluster head of one cluster and a cluster member of the other cluster at the same time. For example, in Figure 3 cluster, the head of cluster C2 is a cluster member of cluster C1. Data transmits from cluster member to cluster head, and at last, all data gathers in the sink node. In the busy period, the cluster header may do some extra works, such as data processing and data aggregation. We will present the details of data transmission in the MAC layer in the next section.

### 4.3. Cluster head rotation

The network deployed for transmission line monitoring is stable. Once dense sets are constructed, they are almost unchanged through the lifetime of the network. Because of intra-cluster energy unbalance, the cluster header consumes much more energy than other members in the same cluster. To balance the energy consumption of nodes and prolong the lifetime of the network, we need to change cluster header at an appropriate time. In our cluster head rotation algorithm, cluster head candidates are selected from the same dense set as the current cluster header belongs to. In this way, we can reduce the overhead caused by exchanging battery information and ensure connectivity of the network at the same time. Cluster head rotation can be triggered in two situation: (i) cluster head fails; (ii) other members’ residual energy of the dense set is \( \alpha(0 < \alpha < 1) \) times greater than residual energy of current cluster head.

Procedures of cluster rotation are as follows:

1. In the busy period before data collection, cluster heads broadcast their residual energy information to the dense set that they belong to, and this process also wakes up the cluster member from its idle period.

2. If a node detects that its residual energy is \( \alpha \) times more than the current cluster head of the same dense set, then it broadcasts a cluster head rotation request in the dense set. Every node sends its residual energy to the current cluster head after receiving the request. After the current cluster head has collected residual
In our scheme, addresses have a topology meaning. We can estimate nodes’ positions by the topology meaning of its address. In this way, addresses are not just identifiers but also indicators of the nodes’ locations in the network. Instead of depending on cluster header/router devices such as CSKIP, our scheme depends on the dense set. Nodes find a packet’s next hop by its address. Therefore, the routing protocol based on the address assignment scheme does not need a routing table or massive route maintenance.

In our address assignment scheme, the logical address of each node consists of three fields: ERN_ADDR, DENSET_ADDR, and BASIC_ADDR. The ERN_ADDR field holds the address of the sink node, which is always an ERN. Nodes with the same ERN_ADDR deliver their sensing data to the same sink node. The DENSET_ADDR field holds the address of the dense set in which the node belongs. Nodes with the same ERN_ADDR and DENSET_ADDR are in the same dense set and are likely to be attached to the same support/tower. The BASIC_ADDR field is used to identify the members of the same dense set.

5. LOCATION-AWARE ADDRESS ASSIGNMENT SCHEME AND ADDRESS-BASED ROUTING PROTOCOL

In some protocols using clustering, such as HSR [61], the node address is defined by a sequence of cluster head identifiers, which result in long addresses, as the level increases. Once a cluster head failed or changed, many nodes’ addresses have to be reallocated, which causes massive overhead. Dynamic Address Routing [13] proposes dynamic addressing mechanisms to cope with the node mobility and to achieve scalable AD HOC routing, but the network used for transmission line monitoring is quite stable. Zigbee’s distributed address assignment mechanism is suitable for hierarchical and stable wireless sensor networks, but it is designed under the assumptions that cluster heads/router devices never change their roles during the network’s lifetime.

5.1. Address assignment

In this section, the process of assigning values to different fields of address of each node is discussed.

(1) ERN_ADDR field: Sink nodes/ERNs can be assigned a unique ERN_ADDR manually or by the control center. Figure 4 shows an ERN 1.0.0 with the ERN_ADDR of 1. Other types of nodes set their ERN_ADDR field as the ERN_ADDR value of the nodes around them. If two different ERN_ADDR values are detected, then the ERN_ADDR with the smaller DENSET_ADDR is chosen.

(2) DENSET_ADDR field: DENSET_ADDR of the dense set with ERN is always 0. The dense set 0 divided the transmission line into two parts, “left” and “right.” Whereas the DENSET_ADDR values of dense sets in the “right” side are assigned even num-

![Figure 4. Illustration of the addressing scheme and address-based routing.](image-url)
bers in the ascending order, the DENSET_ADDR values of dense sets in the “left” side are assigned odd numbers in the ascending order. In this way, the DENSET_ADDR values can indicate the locations of the dense sets in the network.

(3) BASIC_ADDR field: BASIC_ADDR values of ERNS are always 0. BASIC_ADDRs of other types of nodes are assigned from 0 to SET_MEM_NUM-1, where SET_MEM_NUM is the number of members in the dense set the node belongs to.

Figure 4 shows an example of address assignment. For simplicity, the figure only shows the “right” part.

5.2. Address-based routing protocol

Using the address assignment mechanism described in Section 4.2.1, the node’s address indicates its position in the network. To support cluster head rotation, cluster heads have two BASIC_ADDR values. One is obtained from the BASIC_ADDR field assignment, and the other is HEAD_ADDRESS, which is enabled when a node becomes a cluster head and is disabled when it turns into a cluster member. All cluster heads in the network have the same HEAD_ADDRESS, and we can distinguish them by ERN_ADDR and DENSET_ADDR. The address-based routing protocol can be described in two parts: downlink routing and uplink routing. The flow chart of the algorithm is illustrated in Figure 5. In our cluster hierarchy, only cluster heads forward packets; so, we present the routing algorithm in the aspect of the cluster head as follows:

(1) Downlink routing: downlink routing happens when the DENSET_ADDR value of the forwarding packet’s destination is greater than the node’s DENSET_ADDR. If the current node is an ERN, there are two cases: (i) it chooses the cluster head of dense set 1 as the next hop when the packet’s destination address has an odd DENSET_ADDR value and (ii) it chooses the cluster head of dense set 2 as the next hop. If the current node is not an ERN, it chooses the destined node as the next hop if the DENSET_ADDR value of the forwarding packet’s destination is equal to the current_DENSET_ADDR_value + 2. Otherwise, it chooses the cluster head with the DENSET_ADDR value that is the current_DENSET_ADDR_value + 2 as the next hop.

(2) Uplink routing: uplink routing happens when the DENSET_ADDR value of the forwarding packet’s destination is smaller than node’s DENSET_ADDR. If the difference between DENSET_ADDR value of current cluster header (current_DENSET_ADDR_value) and DENSET_ADDR value of destination is greater than 2, the cluster header with the DENSET_ADDR value that is the current_DENSET_ADDR_value -2 is chosen as the next hop. However, if the difference between the DENSET_ADDR value of current cluster header (current_DENSET_ADDR_value) and DENSET_ADDR value of destination is 1 or 2, the destined node is chosen as the next hop.

ERNs collect data and transmit them to the control center by ERNs by cellular modules. The routing algorithm between ERNs and the control center is beyond the scope of this paper.
6. HYBRID MAC

Network utilized in transmission line monitoring has an application-specific topology and traffic pattern. Briefly, we have to overcome the following set of challenges that arise from the characteristics of the topology and traffic pattern:

1. How to adapt to the traffic variability meanwhile achieving energy efficiency and guaranteeing real-time performance? As we mentioned before, the traffic of the network exhibits significant variability and shows a substantial difference from the idle period and the busy period, and the requirements in different periods are also different.

2. How does one avoid the hidden terminal problem in a linear network? The radio’s communication range is designed to cover adjacent support to save energy. Therefore, it is likely to emerge with the hidden terminal problem, especially in the busy period due to its heavy traffic. Figure 6 shows an example: Node A and node C can each communicate with node B, but they are hidden from each other. When node A is communicating with node B while node C is in transmission, a collision occurs. An RTS/CTS mechanism can solve this problem. However, the overhead of RTS/CTS mechanism is massive, especially when data packets are short, and a lightweight scheme is needed to avoid the hidden terminal problem in the busy period.

3. How does one make a tradeoff between energy efficiency and real-time performance in terms of a node’s role in the network? There are two kinds of nodes in the cluster hierarchy: cluster head and cluster member. Only the cluster head forwards packets, and thus, the MAC delay of the cluster head has a significant effect on the packet’s end-to-end delay. Although the MAC delay of a cluster member only impacts the transmission between a cluster member and a cluster head, the cluster member can be more energy efficient with greater MAC delay. With cluster rotation, the roles of the nodes can be changed, and this requires the MAC layer to make a trade-off between energy efficiency and real-time performance in terms of a node’s role.

In H-MAC, nodes utilize a synchronized XMAC during the idle period, and the cluster head and cluster member employ different MAC parameters to meet different requirements; a pipelined TDMA MAC schedule can avoid hidden terminal problem in busy period. Using different media access strategies in different periods, our MAC protocol can achieve reliable transmission in the busy period while not only meeting the real-time requirement but also efficiently saving energy in idle time.

6.1. Pipelined TDMA MAC in busy period

In the busy period, as traffic increases, the hidden terminal problem becomes worse, and this generates lots of collisions and decreases the reliability of the network. The TDMA mechanism works well when traffic of the network is heavy because it can avoid collisions. In the linear network, we adopt a pipelined TDMA schedule in the busy period for two reasons. First, BSNs and DRNs have limited data buffering due to their cost. A pipelined TDMA schedule can reduce the probability of packet loss caused by limited buffering because there is always a transmission opportunity after receiving a packet. Secondly, a pipelined TDMA schedule can reduce congestion and improve throughput. We define the interval between a node’s two serial transmission opportunities (slots) as pipeline period $T_p$. The lower the $T_p$ is, the greater the network’s parallelity and the higher throughput is. A node’s transmission rate (i.e., transmission frequency) can be given by

$$r = \frac{1}{T_p}$$

From the paper [62], for a linear topology, the maximum transmission rate (i.e., transmission frequency) for a node $N$ hops away with interference range $I$ in terms of the number of hops, which is given by

$$r_{\text{max}} = \frac{1}{\text{min}(N, 2 + I)}$$

In this equation, there is a variable range of interference $I$: a node’s transmission interferes with the reception of all nodes that hop away. To achieve the maximum transmission rate,

$$T_p = \text{min}(N, 2 + I)$$

The similar results of the aforementioned equation are also presented in [63–66].

In layer 1, a node’s communication range is between 400 and 800 m, and this just covers adjacent supports; therefore, the interference range $I$ is 1. From the aforementioned equation, we can find that the maximum throughput can be achieved when $T_p$ equals 3. Therefore, the super frame of the pipelined MAC should consists of three time slots at least.
According to the pipelined slot allocation scheme, the whole MAC protocol in the busy period can be divided into two phases as follows:

1. Intra-cluster data collection phase: In this phase, the sensing data of cluster members is collected in the cluster head. This phase needs three time slots to avoid the hidden terminal as analyzed previously. Cluster members utilize the Carrier sense multiple access with collision avoidance (CSMA/CA) scheme to send their sensing data to the cluster head. Within a time slot, nodes contending for the wireless channel belong to the same cluster and can hear each other. Therefore, no hidden terminals exist in this situation. The time slot used for data collection is longer than that used in data forwarding.

2. Data forwarding phase: Data forwarding is carried out between cluster headers. In this phase maximum transmission rate can be achieved by a super frame consisting of three time slots. In the super frame, three time slots are repeated as shown in Figure 7. By this scheme, when a cluster header is in transmission, another cluster header within two hops will not transmit at the same time, and thus the hidden terminal problem is avoided.

By avoiding the hidden terminal problem, the pipelined TDMA MAC protocol achieves high reliability and good throughput at the same time. It also prevents packets from becoming congested in some nodes, and this is imported when nodes have limited buffers to store packets.

6.2. S-XMAC in idle period

In the idle period, communication rarely exists and energy consumption must be considered a major factor in MAC design. At the same time, the aperiodic data is commonly generated by alarms, which also have a real-time requirement. To achieve energy efficiency, a low duty cycle MAC is a widely used solution for a wireless sensor network. This type of MAC protocol can be divided into synchronous MAC protocols and asynchronous MAC protocols. Synchronous MAC protocols need a shorter time (guard time) before sending a packet, but they need to exchange scheduling information between nodes. For example, nodes in Sensor MAC (SMAC) [67] and Timeout MAC (TMAC) [68] send time-synchronization packets every 15 s, and this results in a significant energy overhead [69]. However, in transmission line monitoring, data transmission is rare in the idle period. In this situation, the cost caused by controlled information is high, and the implementation of a synchronous MAC is more complicated compared with an asynchronous MAC. Although asynchronous duty cycle MAC protocols shift the burden of synchronization to the sender, the sender needs to transmit longer preambles to wake up receivers [70].

6.2.1. XMAC.

Our MAC protocol in the idle period is based upon an asynchronous MAC–XMAC [71], as shown in Figure 8. The process of XMAC can be described as follows:

1. Every node wakes up and sleeps periodically. After receiving a data packet, a node wakes up and sends a preamble embedded with a target ID. Then, it waits for an ACK response from the target node. This operation will repeat until the node receives an ACK.
response, and then, the node starts data transmission. The maximum duration of preamble sending is a duty cycle.

(2) When the target node is awake, it sends an ACK response to inform the sender to start data transmission after receiving a preamble embedded with its own address.

(3) If a non-target node wakes up and receives a preamble with a target ID that does not match the node’s address, it returns to sleep immediately and continues its duty cycling.

XMAC embeds a target ID in the preamble. Unlike other MAC protocols with a long preamble, in XMAC non-target receivers return to sleep immediately after receiving a preamble so that overhearing is avoided. The receiver sends a preamble ACK to prevent the sender from sending preambles, and this saves the sender’s energy and improves the real-time performance. In addition, XMAC’s short strobed preamble can be supported by streaming radios and packetizing radios.

The real-time performance of XMAC is dominated by the receiver’s duty cycle. By changing MAC protocol’s duty cycle as in paper [72], differential performances can be achieved. Nodes with a lower duty cycle wake up and listen to the channel more frequently. Therefore, packets can be forwarded in a more timely manner. In our application, traffic in the idle period is very light; therefore, the node’s duty cycle has a larger value (500 ms or 1 s for example) to save energy. On the other hand, only cluster heads forward packets; therefore, we set a higher duty cycle to cluster heads to guarantee the real-time performance. A higher duty cycle is adopted in cluster heads to ensure real-time performance, whereas a lower duty cycle is adopted in cluster members to save energy.

6.2.2. S-XMAC.

No synchronization exists in XMAC to align the node’s duty cycle to reduce long preambles because XMAC is optimized for general cases. However, in the application of transmission line monitoring, we can use loose synchronization to reduce the long preambles. First, because of infrastructure constraints, the network has a linear topology and cluster hierarchy, which makes nodes choose a synchronization source easily. Second, the network collects sensing data periodically, and nodes can utilize such operations to synchronize each other.

In synchronous XMAC, the sender needs a shorter preamble to wake up the receiver because the sender knows approximately when the receiver wakes up. However, because the period of data gathering is about 15 to 30 min, synchronization with this interval may cause significant synchronization errors.

Let $T_{data}$ be the data gathering period of the application. $F_{hc}$ denotes the frequency of the crystal oscillator, and $F_{ppm}$ denotes the frequency deviation of the crystal oscillator. The 32 768 Hz crystal oscillator is widely used in wireless sensor network. Maximum synchronization error can be given by

$$\text{T}_{\text{offset, max}} = 2\lambda + 2 \times \frac{T_{\text{data}} \times F_{hc}}{10^6} \times F_{ppm} \times \frac{1}{F_{hc}}$$

In the aforementioned equation, $\lambda$ represents the initial synchronization error after synchronization is carried out. RF modules on the wireless sensor network platform, such as CC2420, support synchronization operations (time stamping, etc.) in hardware, which can make $\lambda$ on the microsecond level. Compared with the offset brought by frequency deviation, $\lambda$ is negligible. Therefore, maximum synchronization error can also be given by the following:
The worst cases in XMAC and S-XMAC are shown in Figure 9. The worst case in XMAC happens when the sender starts to send preambles just after the target receiver wakes up, and in this case, the preamble sending will last a duty cycle. However, in S-XMAC, the worst case happens when the sender’s clock is $T_{offset_{\text{max}}}$ ahead of the standard clock, whereas the receiver’s clock is $T_{offset_{\text{max}}}$ behind the standard clock. In that case, the preamble sending will last $2T_{offset_{\text{max}}}$. Therefore, the maximum preamble sending time in S-XMAC can be given by

$$T_{preamble} = \begin{cases} T_{cycle} & (T_{cycle} \leq 2T_{offset_{\text{max}}}) \\ 2T_{offset_{\text{max}}} & (T_{cycle} > 2T_{offset_{\text{max}}}) \end{cases}$$

We can learn from the aforementioned equation and Figure 9 that when the duty cycle is two times greater than the maximum synchronization error, S-XMAC is more energy efficient than XMAC. The procedure of S-XMAC can be described as follows:

1. Every node records the wake up time of the adjacent dense set, and every member of the same dense set wakes up at the “same” time.
2. Every node wakes up and sleeps periodically according to its wake up time and duty cycle. After receiving a data packet from the upper layer, the node (sender) does not send preambles immediately until $1/2T_{preamble}$ before the wake up time of the destined dense set’s cluster head. Then, the sender starts to send preambles and waits for the preamble ACK response from the target node. After receiving the preamble ACK, the sender starts data transmission.
3. When the target node is awake, it sends an ACK response to inform the sender to start data transmission after receiving a preamble embedded with its own address.
4. If a non-target node wakes up and receives a preamble with a target ID that does not match the node’s address, it returns to the sleep status immediately and continues its duty cycling.

S-XMAC works basically the same as XMAC does, and the main difference is that in S-XMAC, when the data arrives for the sender, the sender waits for an appropriate time before it starts to send preambles rather than send preambles immediately. In this way, S-XMAC can be more energy efficient by sending fewer preambles before data transmission.

### 6.2.3. Location-based wake up time scheduling.

Urgent data in transmission line monitoring often has higher priority and real-time requirements. It is often an alarm generated by sensors when abnormal events are detected. Urgent data is always upstream (from BSNs to the sink node). However, most downstream data during the idle period is user configuration data and user query data. This type of data also has a looser real-time requirement than urgent data. To make the network support urgent data effectively, we use the geographical location information of nodes in the network to wake up nodes in order, and this makes the upstream data have a better real-time performance. The scheduling scheme is illustrated in Figure 10. From the figure, we can see that the interval of the wake up time of the nodes in the adjacent dense set $T_{schedule}$ should satisfy

$$T_{schedule} > \delta + T_{offset_{\text{max}}}$$

where $\delta$ represents the time a node needs to receive data, which includes maximum data receiving time, data acknowledge time, and dwell time. If $T_{schedule}$ does not satisfy this requirement, as the synchronization error between
receiver and sender increases over time, the receiver may wake up before the sender transmits the preambles, which causes communication failure. In the location-aware address assignment scheme, DENSET_ADDR can indicate the position of the node. So the node’s wake up time can be given by

\[
T_{\text{wake}} = n \times T_{\text{cycle}} - \left( (\text{DENSET_ADDR}+1) \times 2 \right) \times T_{\text{schedule}}
\]

\[
\times \left( n = \left\lceil \frac{((\text{DENSET_ADDR}+1) \times 2) \times T_{\text{schedule}}}{T_{\text{cycle}}} \right\rceil \right)
\]

Because traffic in the network during the idle period is light, collision and packet loss are not taken into account in the following computation.

The average delays of upstream and downstream data are equal in random wake-up time scheduling. The average delay is defined as the time that begins from the moment when data is generated by sensors to the moment when data reaches the sink node. It can be expressed as follows when the hop count is greater than 1.

\[
E(T_{\text{delay}}) = \frac{1}{2} T_{\text{cycle}} + \frac{1}{2} T_{\text{preamble}} + T_{\text{data,tx}}
\]

In S-XMAC, when data arrives for the sender, the sender waits for \( T_{\text{wait}} \) before it starts to send preambles.

When the random wake-up time in XMAC scheduling is used,

\[
E(T_{\text{delay}})_{\text{random}} = \frac{1}{2} T_{\text{cycle}}
\]

When location-based wake up time scheduling is used, the average waiting time of upstream and downstream can be expressed as follows.

\[
E(T_{\text{delay}})_{\text{uplink}} = \delta
\]

\[
E(T_{\text{delay}})_{\text{downlink}} = T_{\text{cycle}} - \delta - T_{\text{preamble}}
\]

In summary,

\[
E(T_{\text{delay}})_{\text{random}} = \frac{N_{\text{hops}} - 1}{2} \left( T_{\text{cycle}} + T_{\text{preamble}} \right) + \frac{N_{\text{hops}}}{2} \times T_{\text{data,tx}}
\]

\[
E(T_{\text{delay}})_{\text{uplink}} = \left( N_{\text{hops}} - 1 \right) \delta + \frac{N_{\text{hops}}}{2} T_{\text{preamble}} + N_{\text{hops}} \times T_{\text{data,tx}} + \frac{1}{2} T_{\text{cycle}}
\]

Using wake up time scheduling, real-time performance of upstream can be improved as follow.

\[
\Delta = \left( N_{\text{hops}} - 1 \right) \left( \frac{1}{2} T_{\text{cycle}} - T_{\text{preamble}} \right) - \delta
\]

7. SIMULATIONS

We use the Optimized Network Engineering Tool (OPNET) to evaluate the performances of our framework.
In our simulations, the simulation area is $4500 \times 50$ m. There are 10 clusters (11 dense sets) in this area. If not mentioned particularly, there are six nodes in each dense set and 66 nodes totally in our network. Nodes in a dense set distribute in an area of $20 \times 20$ m, and the distance between two adjacent dense sets is approximately 400 m. The sink node is in position (20 m, 10 m). The topology of our network is shown in Figure 11.

In our simulations except the sensor’s sensing data which is likely to be less than 10 bytes per monitoring cycle, the sensor type, timestamp of the data may be also needed in application layer. In order to guarantee enough room to contain all these information, we generate packets with length of 20 bytes in application layer. If not specially mentioned the parameters of traffic generated by BSNs are listed in the succeeding text:

<table>
<thead>
<tr>
<th>Periodic data</th>
<th>Aperiodic data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data interarrival time(s)</td>
<td>Constant(1200)</td>
</tr>
<tr>
<td>Data length(byte)</td>
<td>Constant(20)</td>
</tr>
</tbody>
</table>

In our simulations, the simulation area is 4500 × 50 m. There are 10 clusters (11 dense sets) in this area. If not mentioned particularly, there are six nodes in each dense set and 66 nodes totally in our network. Nodes in a dense set distribute in an area of $20 \times 20$ m, and the distance between two adjacent dense sets is approximately 400 m. The sink node is in position (20 m, 10 m). The topology of our network is shown in Figure 11.

In our simulations except the sensor’s sensing data which is likely to be less than 10 bytes per monitoring cycle, the sensor type, timestamp of the data may be also needed in application layer. In order to guarantee enough room to contain all these information, we generate packets with length of 20 bytes in application layer. If not specially mentioned the parameters of traffic generated by BSNs are listed in table II.

The energy consumption model is listed in Table III. The model is based upon measurements of the wireless sensor mote (SiA2445) developed by our lab. SiA2445 [73] uses the power amplifier to achieve longer transmission range, and the transmission range of SiA2445 can reach 600 m in LOS (line of sight) environment, which is suitable for transmission line monitoring.

Parameters staying unchanged during the simulations are listed in the succeeding text:

<table>
<thead>
<tr>
<th>Simulation parameters</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Battery</td>
<td>$2 \times (1.5 \text{ v}, 2300 \text{ mAH})$</td>
</tr>
<tr>
<td>Default max retry time</td>
<td>3</td>
</tr>
<tr>
<td>Zigbee’s MAC parameters</td>
<td>Default IEEE 802.15.4 MAC parameters</td>
</tr>
<tr>
<td>MAC parameters of</td>
<td>$\text{Listen/Wake up time: 6 ms}$</td>
</tr>
<tr>
<td>XMAC and S-XMAC</td>
<td>Dwell time: 10 ms</td>
</tr>
</tbody>
</table>

Note that XMAC has different listen time values in different implementation or Oss; listen time of XMAC is 20 ms in [71], 6 ms in Tinyos, and 6.25 ms in Contiki. Listen time is set as 6 ms in our simulations.

The idle period is much longer than the busy period. Therefore, saving as much energy as possible in the idle period is critical to prolonging the network’s lifetime. At the same time, the network traffic during the idle period is light, but it has a real-time requirement. Therefore, the two main parameters are compared in our simulations: the end-to-end delay and the energy consumption. First, we investigate the impact of the frequency deviation on the S-XMAC performance. Then, the S-XMAC performance is compared with the performance of other MAC protocols, such as XMAC and Zigbee in the idle period.

Figure 12 shows the practical and theoretical average end-to-end delay of packets from the 11th dense set to sink node. In these simulations, the duty cycle of cluster heads and cluster members are 500 ms and 1 s, respectively. Each experiment lasts 10 h. From the figure, we can find that the end-to-end delay increases as frequency deviation increases. For example, average end-to-end delay is decreased by 21.35% when the 10 ppm crystal oscillator is used instead of the 30 ppm crystal oscillator. The reason is that the maximum time offset between nodes ($\text{Toffset}_{\text{max}}$) increases as the frequency deviation increases. Although in the location-based wake up schedule, the wake up timing of a node from an adjacent dense set should be separated to tolerate the time offset; therefore, the wake up timing from adjacent dense sets should be scheduled with a longer interval as $\text{Toffset}_{\text{max}}$ increases. This results in a greater end-to-end delay. The practical and theoretical average end-to-end delay is very close in the figure, which indicates that delay and frequency deviation are proportional as analyzed before.

Figure 13 shows the energy consumption of the cluster head of the second dense set in 4 h with different frequency deviations. Results show that as frequency deviation increases, the energy consumption of the cluster head increases. For example, compared with the energy consumption of the cluster head with 30 and 20 ppm crystal oscillator, the energy consumption of cluster heads with a 10 ppm crystal oscillator is decreased by 41.29% and 25.98%. It is because the transmission power in the transmission is much greater than that in receiving and sleeping states in our wireless sensor platform. With a smaller frequency deviation crystal oscillator, the time offset between nodes becomes smaller, and nodes need fewer preambles to send before sending a data packet, which results in less energy consumption. The cluster members also consume more energy when the frequency deviation of the crystal oscillator is greater, but the difference is not as obvious as with the cluster head. Because a cluster member has significantly fewer packets to transmit, the result is not present because of limited space.

In Figure 14 and 15, we compare the performance of S-XMAC with that of XMAC and MAC protocol of Zigbee non-beacon mode. In Zigbee non-beacon mode, we...
evaluate the performance under the 5 and 10-s data request period of the end device. We evaluate the performance of XMAC and S-XMAC with the duty cycle of cluster heads and non-cluster heads 200 and 600 ms, respectively, and also, the duty cycle of cluster heads and non-cluster heads are 500 ms and 1 s, respectively. The frequency deviation of crystal oscillator in S-XMAC is 20 ppm. Every simulation lasts 10 h.

Figure 14 shows the uplink end-to-end delay of packets from the 11th dense set with a different MAC protocol. In Zigbee mode, the data request period of the end device does not impact the uplink delay. So, we just present the result with a 5-s data request period. From Figure 14, we can find that the Zigbee non-beacon mode achieves the best uplink end-to-end delay. It is because in Zigbee network, routers stay in the working state all the time, and light traffic in the idle period leads to fewer collisions. With the same duty cycle, the uplink end-to-end delay of S-XMAC is much smaller than XMAC. For example, end-to-end delay of S-XMAC can be decreased by 24.94% and 62.44%, respectively, compared with XMAC with the 200 and 500 ms duty cycle of cluster head. This indicates that the location-based wake up schedule scheme of S-XMAC can improve the real-time performance of uplink as we expected. Formula 1 shows that the delay, which is unlikely in XMAC, caused by duty cycle does not accumulate on every hop along the routing path in S-XMAC. When 500 ms instead of 200 ms duty cycle is used, the average

<table>
<thead>
<tr>
<th>Transmit current</th>
<th>Receive current</th>
<th>Idle current</th>
<th>Supply voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>SiA2445</td>
<td>≈ 85 mA @ 3.3 V, +12 dBm</td>
<td>≈ 20 mA @ 3.3 V</td>
<td>&lt; 8 uA</td>
</tr>
</tbody>
</table>

Parameters used in simulations | 85 mA | 20 mA | < 8 uA | 3.3 V |

Table III. The energy consumption model of SiA2445.
uplink end-to-end delay of the 11th dense set just increases 0.1698 s, which is close to the theoretical value 0.15 s. Compared with the uplink traffic, the downlink traffic is even lighter because just ERNs generate downlink traffic. To evaluate the downlink end-to-end delay, we let the ERN generate packets destined to a node of the 11th dense set in Poisson distribution with \( \lambda = 1/300 \). Figure 15 shows the result. When Zigbee non-beacon mode is used, end-to-end delay with the 5 and 10-s data request period is 2.569 and 5.0183 s, respectively. Downlink end-to-end delay is mainly caused by communication between the router and end device, which makes the downlink end-to-end delay close to half of the data request period. The downlink end-to-end delay of S-XMAC with the 200 and 500-ms duty cycle of cluster head is 1.660 and 4.825 s, respectively. Although the downlink end-to-end delay of XMAC with 200 and 500-ms duty cycle of cluster head is 1.582 and 3.915 s, respectively. The downlink delay of S-XMAC is more than XMAC, which is because the location-based wake up schedule is optimized for uplink and degrades the downlink real-time performance at the same time. In transmission line monitoring, the downlink data is less urgent and can tolerate greater delay. Both XMAC and S-XMAC can meet the real-time performance if appropriate protocol parameters are chosen, and S-XMAC is more suitable than XMAC if parameters are the same.

We compare the energy consumption of the cluster head and the cluster member in second dense set. Results are showed in Figure 16. The energy consumption of the cluster head in the Zigbee non-beacon mode is not shown for illustration purposes because it is about 6.4 and 30 times more than that in XMAC and S-XMAC with 500-ms duty cycle, respectively, and the difference further increases as the hops increase; the energy consumes too quickly in the cluster head/router makes Zigbee not suitable for being used in transmission line monitoring without sufficient power supply. In XMAC, with a longer duty cycle, the cluster head may consume more energy, whereas the cluster members consume less energy. This is because XMAC needs to send more preambles before sending data packets; therefore, when the transmission power is much higher than receiving and when the nodes have many packets to send, the energy saved by waking up may not be capable of making up the energy consumed by sending more preambles. In this situation, the node with a longer duty cycle consumes more energy as shown in Figure 16. On the other hand, for nodes with light traffic, a longer duty cycle leads to less energy consumption as shown in Figure 17. In S-XMAC, the number of preambles is not affected by the duty cycle, but it is affected by the frequency deviation of the crystal oscillator. So, with a longer duty cycle, S-XMAC can always save more energy. For example with the 500-ms duty cycle, the energy consumption of the cluster head in second dense set is decreased by 33.8% compared with the scenario with the 200-ms duty cycle. Compared with XMACS, XMAC needs fewer preambles before sending data packets. So, S-XMAC can always out-perform XMAC regardless of the traffic of nodes in energy consumption.

We evaluate H-MAC in the busy period under two different traffic configurations. Under the first configuration, every node generates data with a 20-byte packet length to simulate the traffic generated by sensors such as temperature sensors. Under the second configuration, most nodes generate traffic under the first configuration, a few nodes with sensors such as magnetic field sensors for power quality graph and accelerometers for line galloping monitoring generate data with big volume (50 packets with 100-byte length each collection). We call this type of sensor a bigsensor. In this simulation, the slot length is 40 ms. Each simulation lasts 10 h, and collection starts every 20 min.

To evaluate the reliability of MAC protocol, different numbers of link layer retries are configured, and end-to-end acknowledgement is disabled in the transport layer. When IEEE 802.15.4 MAC is used, there are many redundant packets caused by the hidden terminal problem. We cope with this problem by using a buffer to record the packets that have been received in the MAC layer lately and by dropping received packets if they are already recorded in the buffer. Although in the pipelined MAC, this mechanism...
is not needed because pipelined MAC can avoid the hidden terminal problem. We do not evaluate the energy consumption in this simulation because the busy period is too short compared with the idle period. The energy consumption saved in the busy period can not prolong the lifetime of the network that much.

Figure 18 shows the uplink end-to-end delay of packets from the 11th dense set to sink node with or without a big sensor in the 11th dense set. Results show that with the same traffic configuration, CSMA/CA of IEEE 802.15.4 outperforms pipelined MAC in end-to-end delay. This is so because in pipelined MAC, cluster heads do not forward packets immediately, but they wait until the time slot assigned for sending, which is coming. When the bigsensor is deployed, the end-to-end delay increases, especially in the CSMA/CA MAC layer because collisions increase rapidly. Both CSMA/CA MAC and pipelined MAC can satisfy the application requirements of being real-time because the maintenance information is not as urgent as the alarm in the idle period.

The most concerning performance in the busy period is reliability because in the busy period, the heavy traffic leads to collision and packet loss. The results are shown in Figure 19. When no bigsensor exists in the network, the reliability of CSMA/CA MAC is as acceptable as pipelined MAC when the maximum retry time is 5. When a bigsensor exists, the packet success rate of CSMA/CA MAC is decreased by 14% with the maximum link retry time being at 3% and 6% with maximum link retry time is 5 because the hidden terminal problem becomes worse. As shown in Figure 20, the more bigsensors are deployed, the more the packet success rate of CSMA/CA MAC decreases. Pipelined MAC achieves a high packet success rate. Even though bigsensors are deployed, the results indicate that the pipelined MAC can avoid the hidden terminal problem, and pipelined MAC is capable of supporting different types of sensors in transmission line monitoring.

1816
Figure 21 represents the energy consumption of the cluster head and a cluster member of second dense set. The result shows that the intra-cluster energy consumption is unbalanced in transmission line monitoring. When each dense set has five members, energy consumption of the cluster head is 3.99 times that of the cluster members. When each dense set has nine members, energy consumption of the cluster head is 5.84 times that of the cluster member. Results indicate that cluster heads are bottlenecks of the network’s lifetime. With cluster head rotation, every dense set member has opportunities to become a cluster head, which balances the node’s energy consumption and extends the network’s lifetime. In this simulation, we define the network’s lifetime as the period when the simulation begins to when any node’s energy drops below 50 J. The network’s lifetime via the number of nodes in every dense set is shown in Figure 22. First of all, cluster head rotation can prolong the network’s lifetime as expected. When cluster head rotation is used, the network’s lifetime increases slightly. Results also indicate that we can alleviate the funnel effect and prolong the network’s lifetime by deploying DRNs on supports close to the sink node.

Results of this simulation show the lifetime of the networks cannot meet the requirement of transmission line monitoring. There are two approaches to deal with it. The first is by using a high-capacity battery rather than a 2300mAhx1.5V battery. Second, one can also adjust the S-XMAC parameters to save more energy. The network consumes the most energy in idle periods; so, we can configure the networks with a longer duty cycle and guarantee the real-time performance.

8. CONCLUSION

Transmission line monitoring is important for security and reliability in smart grid. A wireless sensor network is a promising technology for transmission line monitoring due to its low cost, easy installation, large-scale coverage, and fault tolerance characteristics. In this paper, we propose an efficient wireless sensor network framework for transmission line monitoring. In the MAC layer, we propose H-MAC based upon the traffic’s characteristics in the application. In the idle period, we use S-XMAC, which takes advantage of loose synchronization to reduce the number of preambles sent before data packet transmission. We also propose a location-based wake up schedule to optimize the real-time performance of uplink data transmission. In the busy period, we use pipelined MAC to deal with the hidden terminal problem and support ‘‘big sensors.’’ Upon H-MAC, we propose a dense set-based clustering algorithm to resolve the problem of intra-cluster energy unbalance. We also take advantage of the linear network topology to design a location-aware node naming scheme and routing protocols. Simulation shows that H-MAC can meet application requirements in energy saving and real-time performance, and the cluster algorithm can balance the energy consumption all over the network and prolong the lifetime of the network.

ACKNOWLEDGEMENTS

This work was supported by the Natural Science Foundation of China under contacts 61174026, 61233007 and 61374200, Strategic Priority Research Program of the Chinese Academy of Sciences under contact XDA06021000, and Cross-disciplinary Collaborative Teams Program for Science, Technology and Innovation of Chinese Academy of Sciences (Network and System Technologies for Safety Monitoring and Information Interacting in Smart Grid). I wish to thank Prof. Hamid Sharif of the University of Nebraska-Lincoln (UNL) for his advice and assistance with this paper.

REFERENCES


**AUTHORS’ BIOGRAPHIES**

**Junru Lin** received the BE degree from the University of Science and Technology of China, Hefei, China, in 2007, and the PhD degree from the University of Chinese Academy of Sciences, Beijing, China, in 2013. His research interests include wireless sensor networks and industrial wireless communication.

**Baohui Zhu** received the BS degree from Northeastern University, Shenyang, China, in 2010, and the MS degree from the University of Chinese Academy of Sciences, Beijing, China, in 2013. His research interests lie in wireless sensor networks, smart power grids, and embedded systems.

**Peng Zeng** is a professor and the director of the Laboratory of Industrial Control Network and System, Shenyang Institute of Automation, Chinese Academy of Sciences. He received the PhD degree in mechatronics from Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, in 2005. His research interests include industrial automation and the wireless networks for such automation. He also holds the positions of Member, Expert Group of IEC TC65 WG16; Expert, SP100 Standard Committee; Member, Expert Group of FF Wireless Tech. Working Group; and Expert, National Drafting Standard on Wireless Communication Technology for Measurement and Control. His most recent role is to preside over the Important National Science and Technology Specific Project on Industrial Networks.
Wei Liang is currently serving as a professor of Shenyang Institute of Automation (SIA), Chinese Academy of Sciences (CAS). She received the Ph.D. degree in mechatronic engineering from SIA, CAS in 2002. Her research interests are in the areas of wireless sensor network, industrial wireless network, and dynamic scheduling theory. As a primary participant, he developed the WIA-PA standard for industrial wireless networks, which is specified by IEC 62601 and Chinese GB/T 26790.1-2011. As an expert of industrial wireless network technology and standard, she is a member of ISA, IEC TC65 WG16, and WG17. And she also is a member of IEEE and ACM. She is the council member of Digitalcity Measurement & Control Technology TC of China Instrument and Control Society. She is also the council members of China Computer Federation Technical Committee on Sensor Network Instrument and Control Society. She served as peer-reviewer in automation for the National Natural Sciences Foundation of China.

Haibin Yu has been with the Shenyang Institute of Automation (SIA), Chinese Academy of Sciences (CAS) since April 1993, and currently, he is the director of SIA. He received his PhD degree in control theory and control engineering from the Northeastern University in 1997. Prof. Yu’s primary research interests are in industrial communication and control systems, and the development of industrial sensor network technology and system. He is recognized worldwide for his contributions in fieldbus technologies and has recently been elected as an ISA Fellow. Prof. Yu has also been very active in serving professional communities. He is the deputy editor-in-chief of Information and Control. He is also the vice-chairman of China National Technical Committee for Automation Systems and Integration Standardization, and the council members of Chinese Association of Automation and China Instrument and Control Society. He is also one of the expert members for the Intelligent Manufacturing Fund of the Ministry of Science and Technology of the People’s Republic of China. He served as a peer-reviewer in automation for the National Natural Sciences Foundation of China.

Yang Xiao currently is a full professor of the Department of Computer Science at The University of Alabama. He was a voting member of IEEE 802.11 Working Group from 2001 to 2004. He served/serves as a panelist for the US National Science Foundation (NSF), GENI, Canada Foundation for Innovation (CFI)’s Telecommunications expert committee, and the American Institute of Biological Sciences (AIBS), and a referee/reviewer for many national and international funding agencies. His research areas are security and communications/networks. He has published more than 200 SCI-indexed journal papers (including 50 IEEE/ACM transactions papers) and over 200 EI-indexed refereed conference papers and book chapters related to these research areas. Dr. Xiao’s research has been supported by the US National Science Foundation (NSF), US Army Research, The Global Environment for Network Innovations (GENI), Fleet Industrial Supply Center-San Diego (FISCSD), FIATECH, and The University of Alabama’s Research Grants Committee. He currently serves as the editor-in-chief for the International Journal of Security and Networks (JSN) and the International Journal of Sensor Networks (JSNet) (SCI-index).