Data Networking for Autonomous Fatigue Crack Detection

Jinhwan Jung, Deawoo Kim, Hankyeol Lee, and Yung Yi

Abstract One of the useful applications of wireless sensor networks is structural health monitoring, where sensors are distributed to monitor buildings, bridges, large dams, and etc. Out of a large number of application domains we focus on the fatigue crack detection of a structure, e.g., bridge. In this chapter, we summarize the required components for data networking in autonomous fatigue crack detection and explore the design choices there. We first discuss the unique characteristics in delivering data stemming from autonomous fatigue crack detection such as data traffic pattern, network topology, and the necessary degree of performance metrics, e.g., energy efficiency and latency. From the data networking perspective, we present and compare the strength and weakness of various design choices in wireless sensor networks, covering multiple layers in a networking protocol stack.

Key words: Wireless Sensor Network, Low Duty Cycle, Wake-up Radio, Fatigue Crack Detection

Jinhwan Jung
Electrical Engineering, KAIST,e-mail: `jhjung@lanada.kaist.ac.kr`

Deawoo Kim
Electrical Engineering, KAIST, e-mail: `dwkim@lanada.kaist.ac.kr`

Hankyeol Lee
Electrical Engineering, KAIST, e-mail: `mrrays@kaist.ac.kr`

Yung Yi
Electrical Engineering, KAIST, e-mail: `yiyung@kaist.edu`

1 Introduction

1.1 Fatigue Crack Detection

Structural health monitoring (SHM) is undoubtedly a typical one among many applications of wireless sensor networks [4, 12]. Structural health monitoring is conducted by estimating the state of structural health or detecting the changes in a structure. Example applications include a system that monitors wind power[1]. There exists a big project in Europe, which is the EU science and technology key project FP7, named " Health Monitoring of Offshore Wind Farms." Clearly, nondestructive testing using sensors, often forming a large scale wireless sensor network, would significantly cut down the maintenance cost of wind power, lengthen its lifetime, and guarantee its safety.

In monitoring the health of a structure, two major items are investigated and reported: (i) time-scale of change and (ii) severity of change, where time-scale is how quickly the change occurs, and severity is the degree of change. Sensors are responsible for measuring these two items and transferring them to a decision making server. When a structure is large, sensors form a large-scale wireless sensor network, and their generated data are delivered over a multi-hop path.

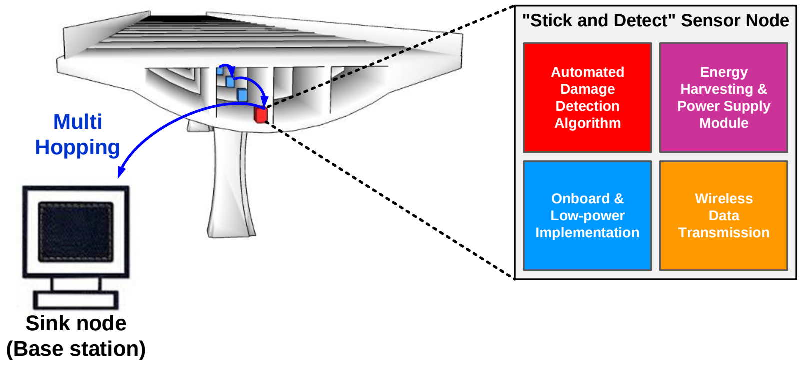
"Stick and Detect" Sensor Node
Automated Damage Detection Algorithm
Energy Harvesting & Power Supply Module
Onboard & Low-power Implementation
Wireless Data Transmission
Multi Hopping
Sink node (Base station)

Fig. 1 Automatic Fatigue Detection System in a bridge

Our focus in this chapter is on the sensors which detect fatigues of a bridge in an autonomous manner, as shown in Fig. 1. Each detection sensor, which we consider throughout this chapter, is located in the bottom side of a bridge road, broadly equipped with (i) automated damage detection module, (ii) energy harvesting module, and (iii) wireless data transmission module. As we recall, the previous chapter was devoted to explaining how such detection is possible through a novel damage detection algorithm and how energy is harvested through external energy sources. In this chapter, we present how to deliver sensed results from sensors over a wireless

[1] China wind power capacity is expected to reach 150 million kilowatts

network. As surveyed in the earlier chapter of this book, networking protocols of a particular wireless sensor network highly rely on the target applications. Thus, we summarize the key features of data generated in autonomous crack fatigue system, which will narrow down the choices in designing a sensor network:

- *Extremely low traffic intensity:* It has been well-known that a collapsed bridge tends to show the signal of being damaged long time before the actual collapse. Thus, the major goal of a fatigue crack detection system is to give a damage alarm prior to being in a dangerous state. In that sense, sensing does not have to be very often. This leads to a desirable scenario that it is possible to make the installed sensors semi-permanent with help of energy harvesting by external energy sources. It would suffice to conduct a fatigue crack detection at the time scale of weeks (i.e., from one week to even one month). This feature of very low traffic intensity has high impact on the designed networking protocols, where a smart choice of MAC and routing should be made for large energy-efficiency.
- *Periodic data generation:* In addition to very low offered traffic load, it is also highly periodic, because fatigue crack detection is not performed on an event-basis. Having the assumption of periodic data generation extremely facilitates the design of networking protocols, because a large amount of energy can be saved by periodically turning off the radios over the duration when data is not generated.
- *Non-realtime, but latency matters:* From the features of fatigue crack detection mentioned above, data does not have to be delivered in real-time. However, due to very low duty cycle of the system, it is possible that sensed data may take a lot of time to be finally transferred to a sink, once the network is ill-designed, which is clearly not desirable by users. Thus, some kind of careful design to reduce latency should be considered, e.g., wake-up time scheduling or its joint optimization with MAC.

1.2 Automatic Fatigue Detection: Networking Perspective

1.2.1 MAC (Medium Access Control)

One of the primary mechanisms for achieving low energy consumption is *duty cycling,* where each sensor node periodically cycles between awake and dormant states. In the fatigue crack detection, as mentioned earlier, due to extremely low traffic intensity, most of time can be spent in the dormant state. Controlling when the radio transceiver becomes dormant can be done in various ways, out of which the following two schemes can be considered. First, the radio transceiver wakes up periodically with some pre-specified period, so that two nodes can communicate if they wake up around the same time. Second, we can equip a sensor node with additional radio, called *wake-up radio*, which is active all the time and is responsible for communicating with another wake-up radio and waking up the main radio if needed. We call the first and the second schemes as *duty-cycled MAC* and *wake-*

up radio based MAC (or simply *WuR-based MAC*), respectively, throughout this paper. Each of these schemes has its own challenges and how these challenges are efficiently solved determines its performance and energy efficiency. For example, in duty-cycled MACs, it is possible for two communicating nodes to wake up with their clocks drifted. In WuR-based MACs, wake-up radios constantly waste energy, and thus designing a radio that consumes ultra low power is necessary. Thus, wake-up radio is often called ULP (Ultra Low Power) radio. In this chapter, we use 'wake-up radio' and 'ULP radio' interchangeably throughout this paper. We will discuss more details in Section 2.

1.2.2 Wake-up Scheduling

With duty-cycled MACs, unless data is delivered over many hops under one wake-up cycle, it would take much time for data to ultimately arrive at a sink node. In other words, over each hop a sender has to hold the transmission and wait until the receiver wakes up based on its wake-up schedule. Thus, combating against low latency in duty-cycled WSNs is important and highly challenging, where smart control mechanism should be devised for smaller latency by co-working with MAC and routing. One way of reducing long latency in duty-cycled MACs is by scheduling wake-up times of nodes at slightly different times, rather than waking up at the same time, considering the structure of routing. This idea is often called as pipelining of wake-up times, where the basic idea is to schedule the different wake-up times for different nodes, so as to they are slightly shifted in the sequence of their depths with respect to the target sink node in the routing paths.

1.2.3 Time Synchronization

As will be discussed later, in our fatigue crack detection with very low offered loads, synchronous duty-cycled MACs with wake-up scheduling may be be a good choice for energy-efficiency and latency reduction. Such a design choice naturally needs time synchronization among node. In the design of a time synchronization mechanism in WSNs, there are various factors and challenges that should be considered as summarized next. First, clocks embedded in the processor of a sensor node tend to be made of low-quality crystals. This results in frequent clock drifts accompanied with an considerable amount of clock skew. In order to maintain time synchronization among sensor nodes, relative difference in the reference clock drift and offset must be minimized. Second, the resources in WSNs are highly limited in terms of energy, processing power, memory, and communication bandwidths. Thus, too heavy control overheads for time synchronization may not be allowed. Third, communication environments in WSNs are also very restricted such as frequent transmission failures, channel contentions, and asymmetric message delays. These harsh communication environments prevent the time synchronization in other conventional networks from working as expected, requiring a careful design in WSNs.

2 Medium Access Control

2.1 Two Methods: Duty-cycling vs. Wake-up Radio

As stressed repeatedly throughout this paper, one of the key features of autonomous fatigue crack detection is that the data rate is extremely low. Thus, from the perspective of medium access control, it is of prime importance to design a MAC protocol so as to fully exploit this feature and run the network in an energy-efficient manner. How to control medium access under this environment can be classified into two schemes: (i) duty-cycled MAC and (ii) wake-up radio based MAC (WuR-based MAC), each of which has its own pros and cons, as elaborated next.

Duty-cycled MAC

One of the primary mechanisms for achieving low energy consumption is duty cycling, where each sensor node periodically alternates between awake and dormant states. The key parameters in characterizing a rule of duty cycling include the durations of sleep and dormant states. In the fatigue crack detection, as mentioned earlier, due to extremely low traffic intensity, sensor node would be mostly in the dormant state. In this case, which MAC is appropriate in terms of energy saving? There are two broad families: synchronous and asynchronous. In synchronous MACs, nodes' time is maintained with the (locally or globally) same clock, where in asynchronous MACs, no such time synchronization is conducted. Since traffic is periodically generated, it is intuitive that asynchronous MAC is not a good choice. However, there still exist challenges such as large clock drift while nodes are asleep, the frequency of synchronization, and synchronization overhead.

WuR-based MAC

Rather than being awake and dormant in a periodic manner, one can propose to install another low-energy radio, referred to as *wake-up radio,* and use it to wake up the main transceiver whenever data communication is needed. In other words, wake-up radio corresponds to a "guard" who is always awake but spends very low energy. The notion of wake-up radio completely eliminates the worry about clock drift and simply realizes the "wake-up and go" mechanism. However, the challenges are: how to share the channel between wake-up and main radios, possibly different transmission ranges of two radios, and extra energy of wake-up radio and the added implementation complexity to sensors. Fig. 2 shows a concept of MACs with ultra low power wake-up radios.

Duty-cycled vs. Wake-up Radio

The main question would be which method is more energy efficient for fatigue crack detection? The answer is it depends on the traffic intensity and how energy-efficient

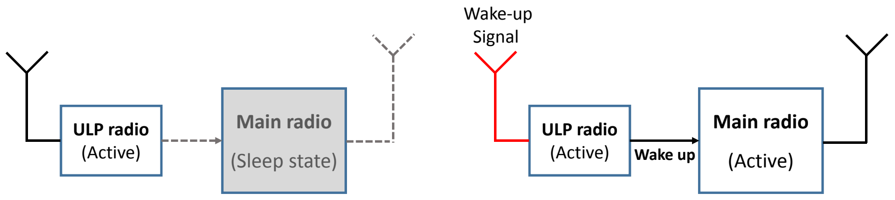

Fig. 2 Concept of Ultra Low Power Wake-up Radio (WuR). Left: the main radio in the sleep state. Right: the main radio in the active state by the wake-up signal from WuR.

the designed duty-cycled MAC is and how much energy is wasted by a wake-up radio. The reason is as follows: Since wake-up radios in WuR-based MACs are always turned on, they constantly use a certain amount of energy, but no complicated MAC-level mechanism to combat against clock drift as done in duty-cycled MACs is needed. Intuitively, when traffic intensity becomes smaller, duty-cycled MACs' energy efficiency becomes better since energy used for tackling clock drift becomes smaller than that by wake-up radios, and vice and versa. This implies the importance of good designs of both types of MACs as well as the significance of adaptive use of either type of MACs.

2.2 *Duty-cycled MAC*

2.2.1 Challenges

In duty-cycled MACs, for maximizing energy-efficiency, it is important to reduce the radio's active time, but the challenges in designing an energy-efficient duty-cycled MAC, especially for extremely low offered load as in the fatigue crack detection application exist, as summarized in what follows.

◦ *Communication coordination:* The first challenge lies in how to make both transmitter and receiver wake up at the same time so that the communication between them becomes feasible. An immediate way of achieving such communication coordination is to let all nodes be programmed with a period of wake-up and sleep. However, such a simple scheme does not work simply because the clock in a sensor tends to be easily adrift. This problem becomes more serious when the duty-cycle is extremely low, e.g., an order of weeks in our fatigue crack detection case. This wake-up time synchronization has long been a critical issue in designing a duty-cycled MAC, leading to two broad classes of protocols: asynchronous vs. synchronous ones. In asynchronous MACs, sensor nodes do not have any wake-up schedule, and the nodes which have packet to send "poll" each of their intended receivers. This is done by sending a beacon for a long time, e.g., [20, 2] or by listening to the receiver's incoming signals, e.g., [23], until the receiver wakes up. In synchronous MACs, e.g., [25], a certain appointment is pre-made, so that they communicate when they wake up around the same time. However, again due to the problem of clock drift, it is required to frequently synchronize nodes' clock to make nodes wake up at the right time. Clock synchronization becomes an overhead that is not possessed by asynchronous MACs. More details will be discussed in Section 2.2.2.

- *Latency:* Although energy-efficiency is one of the primary concerns in WSNs, time-urgency often becomes important in some applications, e.g., fire alarm. In our fatigue crack detection system, time urgency is not a critical issue, but when using a duty-cycled MAC, unless data is delivered over multiple hops under one wake-up duration, it would take a large amount of time for data to arrive at a sink node. In other words, a sender has to hold the transmission and wait until the receiver wakes up based on its schedule. Remember that every sensor wakes up, for example, every three weeks! Also, when data from one sensing is delivered over a large number of wake-up cycles, it becomes another source of energy waste. Note that MAC is just responsible for one-to-one communication. Thus, additional control mechanism should be involved for small latency. One of the possible solutions for this long latency problem is to schedule the wake-up times differently so that multi-hop forwarding can be accomplished over one or a small number of duty cycles. We will discuss this in Section 3.
- *Transmission failure:* The environment of the place where a bridge is constructed is harsh, and thus transmission failures are very common. Similarly to the challenge of latency, these transmission failures may be another source of very long latency, because a poorly-designed MAC may require multiple duty cycles in order to have transmission success between two nodes, if unsuccessful packets are simply retransmitted. We discuss this issue in Section 3.

2.2.2 Asynchronous vs. Synchronous

Asynchronous

In asynchronous MACs, sensor nodes' wake-up period is not synchronized. Thus, whenever a transmitter has a packet to deliver, it detects when its intended receiver is ready for reception and carries out packet delivery. There are two approaches in asynchronous MACs: (i) sender-initiated and (ii) receiver-initiated.

- *Sender-initiated:* In this approach, a sender first expresses its intention of communication via request signal (e.g., preamble, control or even data packet themselves) over a time interval during which its receiver detects whether somebody wants to communicate with itself. X-MAC [2] is a representative protocol.
- *Receiver-initiated:* Different from the sender-initiated approach, in this approach, a receiver expresses its readiness to receive packets to neighboring potential senders. RI-MAC [23] is a typical example of this class.

In this chapter, we briefly explain X-MAC as an example of asynchronous MACs, see Fig. 3. In X-MAC, a sender uses a *strobed* preamble composed of successive

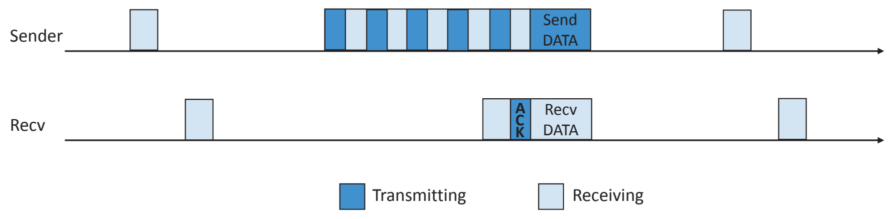

Fig. 3 X-MAC Protocol

packets to denote that it has information to deliver. Each packet contains receiver's address so that its intended receiver can recognize whether it is the sender's target or not. During the repeating state transitions between wake-up and sleep, if the receiver recognizes that it is the target receiver, it stays at the active state and sends ACK as a sign of its readiness. Through this procedure, communication is coordinated and data packet is transmitted by the sender.

Synchronous

In synchronous MACs, all the nodes in a neighborhood have the same time of wake-up period, for which they run a certain synchronization procedure with help of control message signaling for time synchronization. To exemplify, we consider SCP-MAC [26], as depicted in Fig. 4.

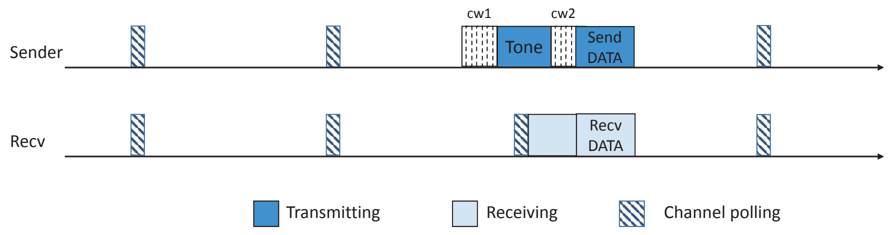

Fig. 4 SCP-MAC Protocol

In SCP-MAC, all nodes maintain local synchronization by exchanging synchronization packets periodically. The amount of clock drift among the corresponding nodes depends on how often time synchronization is conducted. Instead of highly accurate synchronization, SCP-MAC exchanges synchronization packets less frequently than its earlier approach, e.g., S-MAC [25]. SCP-MAC uses a tone signal to ensure the success of communication coordination; By transmitting the tone signal with the length of possible clock drift, a sender can establish the communication channel despite inaccurate synchronization, where receivers have only to poll the channel periodically. Note that the length of the tone signal relies on the frequency of time synchronization.

Comparison: Extremely Low Offered Loads

The key question here is which is better in terms of energy-efficiency, synchronous or asynchronous MACs. The answer depends on various factors, but the key factor must be the amount of traffic to be processed. We provide a qualitative comparison of both schemes by considering three scenarios in terms of offered load, where our major interest is the case when the load is extremely low.

◦ *High offered load:* In this case, synchronous MACs are more energy efficient than asynchronous ones. The reason is as follows. Since asynchronous MACs works without any pre-scheduled time, it is necessary for a sender to send a long preamble signal prior to each packet transmission. When packets are generated frequently, the overhead of the long preamble tends to exceed that of time synchronization
◦ *Moderately low offered load:* When offered load is reduced to a moderate amount, the situation in the case of high offered load is reversed so that the energy consumption due to time synchronization in synchronous MACs becomes larger than that due to long preambles in asynchronous MACs.
◦ *Extremely low offered load:* As discussed in the above, if we follow the comparative argument of energy wastes due to synchronization and long preambles, as the generated traffic intensity reaches a point where offered load becomes significantly small, probably as in our application of fatigue crack detection, it seems that asynchronous MAC is the winner. However, with the knowledge of extremely low offered traffic load, clock synchronization does not have to be conduced as often as high or moderate offered loads, but at that cost of small clock drift. Thus, one can optimize the frequency of clock synchronization so that synchronous MACs are more energy-efficient than synchronous MACs. In other words, once the frequency of clock synchronization decreases, a tone-signal based solution such as in SCP-MAC [26] deals with the clock drift issue, which can be more effective in terms of energy consumption than a significantly long preamble signal in asynchronous MACs.

We comment that the above comparison just shows a qualitative trend, where what is better when highly depends on the particular design choice, also affected by many other internal and external factors.

2.3 WuR-based MAC

2.3.1 Challenges and Design Choices

Challenges

WuR-based MACs differ from duty-cycled MACs in that senders wake up receivers on-demand, whenever communication is intended by senders. This is done by equipping a sensor node with a secondary radio, which is designed to work at a very low data rate and very low power, thus often called ULP (Ultra Low Power) radio. The following challenges exist in designing a MAC with ULP radio:

- ◦ *ULP radio's energy-efficiency:* Since ULP radio is normally in the always-on state and thus constantly uses battery power, it is desirable to design a ULP radio that consumes a very small amount of energy. Several researches [19, 27, 18] report that their ULP radios consume less than 50μW, as opposed to main radios, e.g., cc2420, that typically consume more than 50mW for communication. The power consumption of a ULP radio depends on other factors such as data rate and signal reception sensitivity, which determines ULP radio's performance. Thus, various factors should be taken into consideration, where the tradeoffs among those factors should be smartly exploited.
- ◦ *Communication range:* As mentioned in the previous paragraph, ULP radios are often designed to have the reception sensitivity that is worse than that of the main radio. This design choice helps in increasing energy-efficiency, but generates inefficiency when sensors are used for multi-hop communication. This is because difference in the reception sensitivity of both radios causes a pair of nodes with capability of communication with their main radios to fail in waking up each other with ULP radios. This complicated simulation due to heterogeneous communication ranges of both radios can be a source of more energy consumption. To avoid this, one often deploys sensors on the assumption that they can communicate only within ULP radio's communication range, which, however, prevents the degree of spatial reuse from being fully exploited, thus leading to inefficiency of operating the resulting network. However, the reception sensitivity of some of the state-of-the-art ULP radio reaches -83dBm [18], being reported to almost catch up with that of the 802.15.4 Zigbee main radio.

Design Choices

Major design choices in WuR-based MACs are two-fold: (a) how to specify a destination node and (b) which channel is used for wake-up radios.

- ***How to Specify Destination Nodes***
  - ◦ *Range-based:* A packet sender transmits the tone-based wake-up signal to its neighbors *all* of which wake up their main radios and wait until the data packet is transmitted. This scheme is useful for multicasting/broadcasting, but for unicasting. For unicast traffic, non-intended receivers waste their energy due to overhearing.
  - ◦ *Identity-based:* The wake-up signal transmitted by a packet sender contains a bit sequence encoded with the target destination address. Thus, only intended receiver wakes up its main radio after decoding the wake-up signal. This scheme achieves better energy efficient than range-based schemes for unicast traffic, but extra energy for sending and receiving the bit sequence signifying the destination address is consumed.

Table 1 Survey of Wake-up Radio based MAC protocols

Protocols	Destination Specification	Channel Usage
Subramanian *et al.* [22]	Range-based	Shared
Song *et al.* [21]	Range-based	Shared
Miller *et al.* [16]	Range-based	Separate
Guo *et al.* [11]	Range-based	Separate
LEEM [7]	Identity-based	Separate
Francisco *et al.* [9]	Identity-based	Separate
RTWAC [1]	Identity-based	Separate
CMAC [5]	Identity-based	Shared

- ***Which Channel for Wake-up Radio***
  - *Shared Channel:* ULP radios share the channel used by main radios. In this design, frequency resource is utilized more flexibly and thus more efficiently, because no exclusive channel access is provided, where the radio, either of ULP radio or main radio, with larger needs can utilize the entire frequency resource without explicit control. However, there may exist contention between two radios.
  - *Separate Single Channel:* A separate channel from that used by main radios is reserved for ULP radios. Then, no contention of frequency resource occurs between two radios, but the efficiency in resource usage becomes lower than that in the design with shared channel, because of explicit channel separation.

2.3.2 Taxonomy and Example Protocols

In this section, we provide a survey of some of MAC protocols of sensors equipped with ULP radios, see [6] for a more exhaustive survey. Table 1 shows the taxonomy of the state-of-the MAC protocols with ULP radios. We henceforth briefly explain two MAC protocols, RTWAC [1] and LEEM [7], both of which are identity-based and use a separate channel for ULP radios.

RTWAC

Radio-triggered Wake-ups with Addressing Capabilities (RTWAC) is a protocol that naturally inherits how it would be appropriate to use a wake-up radio. As shown in Fig. 5, the ULP radios is always in the active state, and transmits wake-up signal whenever there is a need for data transmission. Since it is an identity-based scheme, the wake-up signal includes the target receiver address, and thus the receiver's main radio and microcontroller are woken up on demand.

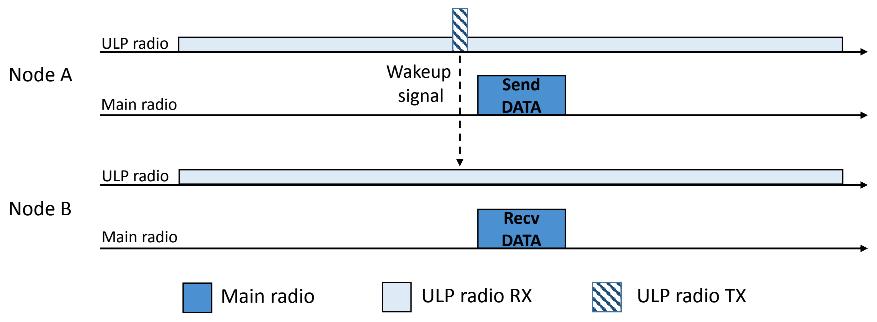

Fig. 5 RTWAC

LEEM

Latency minimized Energy Efficient MAC (LEEM) is a WuR-based MAC protocol that slightly differs from RTWAC in that ULP radios are also periodically duty-cycled for further energy saving, as depicted in Fig. 6. Thus, the wake-up signal can be transmitted only when the wake-up radio is turned on, which, however, can be a source of latency, forming a tradeoff between energy-efficiency and delay. The wake-up radio at the sender side transmits a REQ as a wake-up signal, but it is required for its receiver to send back an ACK in response to the REQ in order to ensure the communication coordination between the wake-up radios, after which data transmission occurs. Additional energy-efficiency is at the cost of more complicated control signaling and extra time synchronization for wake-up radios' communication coordination.

In LEEM, an advanced feature is included for smaller latency in multi-hop data forwarding. That is a *reservation scheme*, corresponding to a kind of *pre-wakeup scheme* that if I am a relay node, then while I perform data communication, I wake up the next-hop node even during the data transmission over the previous hop. Fig. 6 shows such a reservation scheme, where node *B* sends the RES (as a reservation) packet to node *C* while the data transmission occurs between nodes *A* and *B*.

Which Design Choice for Fatigue Crack Detection Sensors?

First, the identity-based scheme seems better, because data communication in this application is mainly unicast-oriented, where waking up all neighbor nodes as in the range-based schemes leads to considerable energy waste. Second, due to the feature of offered load that is significantly small, ULP and main radios are rarely activated.

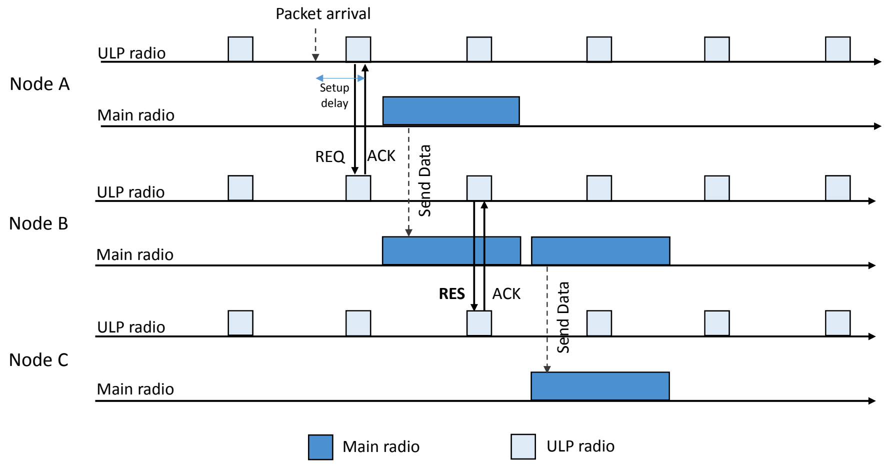

Fig. 6 LEEM

Thus, explicit separation of channels for each radio is unnecessarily complex, where packet collisions between the transmissions over both radios rarely occur.

2.4 Comparison: Duty-cycled vs. Wake-up Radio

Recall that in Section 2.1 we provide a qualitative comparison of duty-cycled MACs and WuR-based MACs. In this section, we present a simple quantitative evaluation of both MACs to address the issue of what is better when in terms of energy-efficiency, probably depending on the offered traffic load. Here, we restrict our attention to the energy-efficiency of only a pair of a transmitter and a receiver. To that end, for a WuR-based MAC, we use RTWAS on the assumption that a sensor is equipped with the ULP radio in [13]. For a duty-cycled MAC, we use the SCP-MAC.

In this comparison, we conduct a numerical study, where the energies in both protocols are modeled appropriately, whose details are omitted due to space limitation. We denote by E_{duty} and E_{ulp} the total amount of energies consumed in the WuR-based MAC and the duty-cycled MAC that we consider here, respectively. Then, E_{duty} and E_{ulp} are given by:

$$E_{duty} = E_{d\_data} + E_{d\_sync} + E_{d\_wait}, \quad (1)$$
$$E_{ulp} = E_{u\_data} + E_{u\_wait}. \quad (2)$$

In the above, E_{duty} consists of (a) energy for data transmission including the energy for sending preamble, E_{d_data} (b) energy for synchronization, E_{d_sync} and (c) energy for periodic listening E_{d_wait}. E_{ulp} is the sum of the energies for data transmission E_{u_data} and for waiting by the ULP radio E_{u_wait}. Note that we ignore the energy when nodes are asleep, because it amount to about 10 thousand times smaller then the amount of other energies in normal main radios, e.g., CC2420 [24].

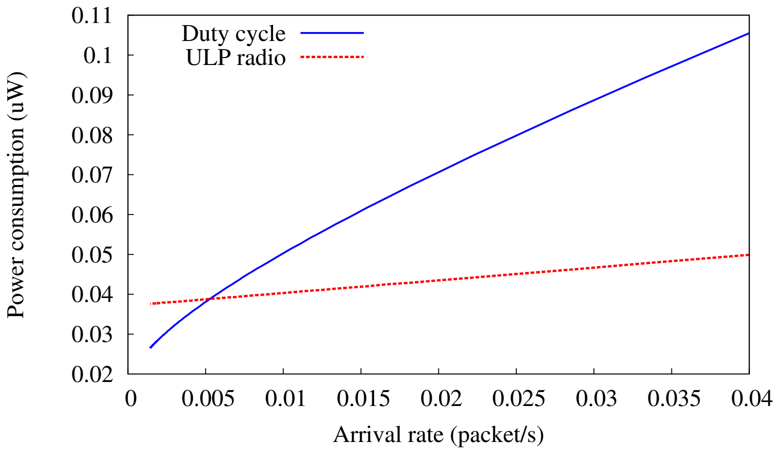

Fig. 7 Comparison of the total energy consumption: Duty-cycled MAC and Wake-up radio based MAC.

Fig. 7 plots the total consumed energies per sec of both schemes as we vary the rate of generated data traffic. To summarize this result, There exists a threshold of arrival rate beyond which the WuR-based MAC outperforms the duty-cycled MAC. This means that for extremely low traffic intensity, duty-cycled MACs are highly likely to be more energy-efficient. We now explain how the results as in Fig. 7 are obtained. In case of many packets to send, the duty-cycled MAC has to send long preambles for every transmission. This energy due to long preambles exceed than that for data transmission (i.e., E_{u_data}) in the WuR-based MAC. However, in case of low offered traffic load, the amount of energy waste due to ceaseless wake-up by the ULP radio becomes dominant in the WuR-based MAC, whereas E_{d_wait} decreases when the offered load decreases. Thus, WuR-based MAC consumes more energy than the duty-cycled MAC. In the application of fatigue crack detection, when a network designer chooses to a data generation interval so as to be smaller than the threshold intensity, it turns out that the duty-cycled MAC is the right choice to deploy. However, note that our analysis here just shows a global trend, and the actual threshold data arrival rate may change, depending on how to design and implement each type of MACs. Also, the total amount of energies of both schemes can also change when we enlarge our view to what happens in the multi-hop transmission from sensors to a sink node.

3 Wake-up Scheduling

3.1 Motivation

Recap: MAC for Light Traffic Load

As mentioned earlier, the key features of the autonomous fatigue crack detection system considered in this chapter are very low offered traffic load that is periodic. Thus it should be operated with extremely low duty cycle, where most nodes are scheduled to be active for a short duration and stay asleep for most of time. Two MACs—duty-cycled MAC and WuR-based MAC—may be able to be deployed under such light traffic load, where we discussed that in a significantly low traffic intensity, duty-cycled MAC can be more energy-efficient. In this section, we address one of the important problems of MACs with low duty cycles, *long latency*, and survey the mechanisms to cope with it.

Problem: Long Latency

In our fatigue crack detection system, time urgency is not a critical issue, but when using a duty-cycled MAC, unless data is delivered over multiple hops under one wake-up duration, it would take very long for sensed data to arrive at a sink node. Recall that it may be sufficient to monitor a target bridge every week or even every month. Also, when data from one sensing is to be delivered over a large number of wake-up cycles, it becomes another source of energy waste.

Wake-up Scheduling

One way of reducing long latency in duty cycled MACs is by appropriately scheduling wake-up times of sensors, rather than synchronizing them so as to wake up at the same time. This idea of wake-up scheduling is often called a pipelining technique, because wake-up times are scheduled to be slightly shifted in the sequence of their depths with respect to the target sink node in the routing paths. Note that by the nature of wake-up scheduling a certain accuracy of time synchronization is needed to enable this scheme. In this section, we introduce three ideas realizing the idea of pipelining in literature: D-MAC [15], Fast Path Algorithm [14], and Multi Scheduling [3].

3.2 D-MAC [15]

D-MAC was proposed as a MAC to realize the idea of pipelined scheduling of wake-up times at the MAC layer. D-MAC runs in an environment, where once the nodes are deployed, they do not have high mobility during a reasonable time and a given routing algorithm forms a notion of data gathering tree. D-MAC schedules the wake-up times of each node based on the depth of the formed data gathering tree. Fig. 8 pictorially shows the concept of the pipeline scheduling of D-MAC.

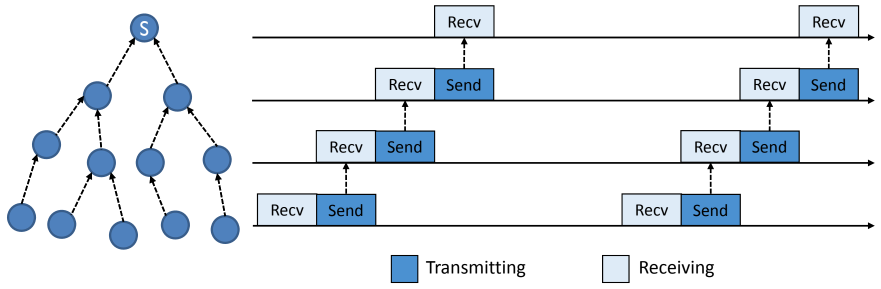

Fig. 8 Pipeline Scheduling in D-MAC

Basic Mechanism

Nodes in the same depth will be scheduled at the same time slot. Therefore, those nodes should compete for the channel when they wake up. To reduce collisions during this contention time, each node at the same depth conducts a random back-off within a contention window at the beginning of a scheduled slot. In this way, D-MAC allows a sequential packet forwarding from sensor nodes to a sink node, so as to reduce end-to-end multi-hop delay.

Contention Losers and multi-packet transmissions

A contention-winning sensor at a given depth in the routing tree is able to send a packet to its parent node. However, the contention-losers in the same depth are not allowed to access the channel and send packets during this duty cycle, naturally resulting in the delivery latency of their packets. To cope with it, D-MAC proposes a scheme called *data prediction* as explained in the following: Once a sensor node receives a packet, it assumes that there are another nodes which have pending packets to transmit. Thus, rather than going to the sleep state until the next duty cycle, the packet-receiving node goes back to the active state just three time slots after the packet reception. The reason for waiting for *three* time slots is that the next node in the multi-hop path (given by the routing tree) would forward the packet in the next three time slots after which the channel is highly likely to be clear. Since all nodes in the path take the same procedure, several packets stored in the queue of node can be forwarded to the sink node in almost one duty cycle.

In the case when there are multiple packets to be forwarded, D-MAC adaptively adjusts its duty cycle according to the given traffic intensity. In D-MAC, a node marks the *more data flag* on its outgoing packet. Once a node receives the packet with *more data flag*, again it wakes up just in three time slots to receive the remaining packets so as to a multiple of packets can be transmitted during one duty cycle.

Pros and Cons

D-MAC improves the end-to-end latency by scheduling wake-up times of nodes differently with help of a tree-based routing. It additionally includes the mechanism of adapting the length of duty cycle to dynamically respond to the offered traffic pattern and to reduce the packet delays of the sensers which lose at channel contentions. However, D-MAC does not specify any way of combating against packet losses due to channel errors, in which case D-MAC would experience severe long sleep latency and thus energy waste.

3.3 Fast Path Algorithm [14]

Basic Mechanism

Fast Path Algorithm (FPA) was also proposed to address the sleep latency problem as a successor of D-MAC. In FPA, all nodes are usually scheduled to simultaneously wake up and act similarly to a synchronous MAC protocol. However, when an end-to-end flow is established from a sensor node to a sink, a notion of fast path is constructed, where the nodes along the established fast path are scheduled to wake up exactly when the previous-hop node is ready to send a packet. The fast path is established by piggybacking a fast-path setup message along with the first message in the direction of data flow. This first message can be either the routing configuration packet or the first data packet in the flow. Once each node receives the fast path setup message, each node estimates a fast-path schedule by using its hop-distance from the sender node. Then, the nodes wake up for an additional amount of time according to its fast-path schedule, and forward their packet to next-hop node. Thus, compared to D-MAC, FPA can be regarded as a routing-independent approach, because wake-up scheduling is constructed on-demand by exchanging signaling message for a fast path.

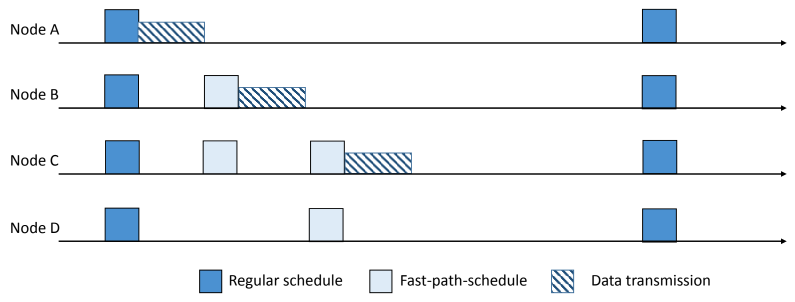

Fig. 9 Fast data fowrarding with fast-path-scheduling

Fig. 9 shows an example of wake-up patterns of nodes in FPA. As explained earlier, each node wakes up based on both regular and fast-path schedules. For example, right after *A* sends a packet to *B*, *C* wakes up to forward the packet from *B*. In this way, the packet can be forwarded sequentially along the fast path without much sleep latency.

Pros and Cons

D-MAC is highly coupled with the data gathering tree from the routing layer, which restricts the type of messages that can be transferred by D-MAC's wake-up scheduling logic. However, FPA is working independently of the routing information, where regular duty cycles are maintained with fast-path being an additional scheduling mechanism. Thus, FPA has more freedom in the kinds of messages that can be communicated. Also, since the fast path is established when the flow is made on demand, it allows the pipelined schedule to adaptively change in response to the topology change. The weakness of FPA is that when the network size is large, and thus the number of flows increases, it becomes less scalable and more complicated, opening possibility of low energy efficiency.

3.4 Robust Multi Scheduling (RMS) [3]

Motivation

The wake-up scheduling described in earlier sections is based on a *single* pipeline. However, this single pipeline scheme works only when the links in the schedule are reliable, whereas in WSNs links' quality often change due to the harsh environment in the deployed areas and thus links often become unreliable for some time duration. In such a lossy link condition, once a link failure happen, a node should wait until the next duty cycle to forward packets. This causes sleep latency and it can be worse when the duty cycle is extremely low. In such a case, the employed routing protocol detects the status of link unreliabilities and reconfigures the routing paths, and thus the wake-up schedule should be also re-configured accordingly. Robust Multi-pipeline Scheduling (RMS) was proposed to tackle such unreliable links by forming multiple pipeline schedules.

Basic Mechanism

Fig. 10 shows a simple example for multi-pipeline scheduling, where the number of each node corresponds to the time slot in which each node is scheduled to wake up. Note that all nodes except a sink node *S* have multiple parents. The node *A* tries to forward a packet to *B* at time slot $\{2\}$. If the link is reliable, it would work similarly to a single pipeline scheme. However, if *A* fails to send the packet to *B*, *A* can try *C* as a next forwarder at time slot $\{3\}$. In this way, each node has several chances to forward a packet in one duty cycle, when link failure happens. Therefore, forwarded packets turn out to experience small sleep latency even in lossy link conditions.

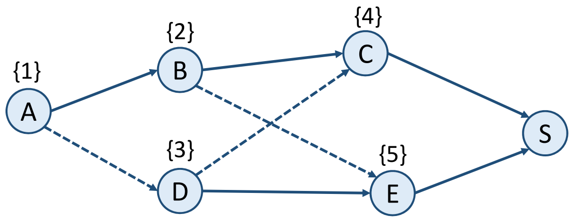

Fig. 10 Multi-pipeline scheduling

To construct a multi-pipeline schedule, maintaining multiple parents nodes for each child and accordingly determining the wake up time slot are important. In RMS, each node selects its parents among its potential forwarders based on the link quality and distance from the sink node. Each node is aware of the wake up schedule of its multiple parents, and for sequential packet forwarding, each node should wake up right before one of its parents nodes' wakes up. Then, each node decides on the forwarder among its parents, which can minimize the expected latency.

Pros and Cons

RMS provides robustness to wake-up scheduling by providing multi-pipeline schedules even in the lossy link conditions. Naturally, it requires the link quality information, but it is often challenging to estimate the link quality with some degree of accuracy, since too heavy estimation becomes another source of energy waste with low-cost transceivers. Clearly, the change of link qualities requires schedule reconfiguration, incurring additional overheads.

4 Time Synchronization

4.1 Why Synchronization and TPSN

In an application with periodic, low offered traffic load such as our autonomous fault crack detection, we have discussed that synchronous duty-cycled MAC in conjunction with pipelined wake-up scheduling can be a good choice of saving significant energy consumption. In realizing such a design choice, of paramount importance is the module of time synchronization. There exist many proposals on how to synchronize time. In this chapter, we introduce TPSN (Time-sync Protocol for Sensor Network) [10] as an example.

Time-sync Protocol for Sensor Networks (TPSN) is a network-wide time synchronization methodology in a wireless sensor network. TPSN is a sender-receiver synchronization designed (i) to decrease receiver-side synchronization complexity, and (ii) to increase synchronization scalability of a sensor network. As will be elaborated later, TPSN consists of the following two phases: (1) *level discovery phase,* where the network-wide tree structure is built from the root node, and (2) *synchronization phase,* where pairwise time synchronization takes place based on the prebuilt hierarchical tree structure.

4.2 TPSN: Protocol Behaviors

Pairwise Time Synchronization

The key module of TPSN is a two-way handshaking mechanism that calculates the propagation delay and the clock drift, based on which a node can by synchronized to another node. Note that such a two-way handshaking is popularly used in other time synchronization protocol such as NTP (Network Time Protocol) [17] in wired networks. However, we will present that just a simple two-way handshaking is not acceptable in wireless sensor networks, but prior to it, in this paragraph we first explain a basic principle of two-way handshaking, as illustrated in Fig. 11.

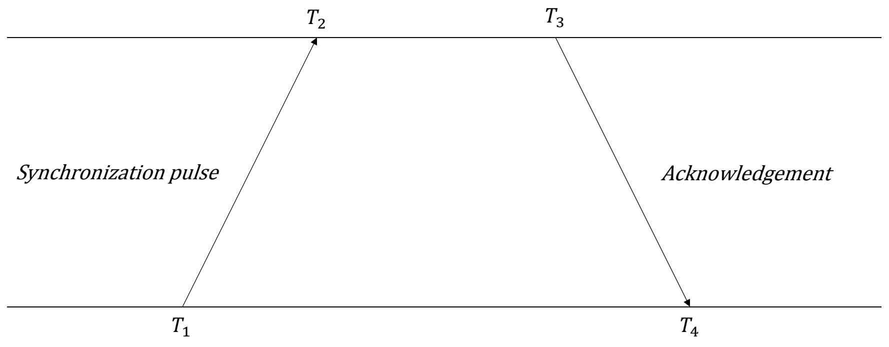

Fig. 11 Two-way handshake of TPSN

The synchronization-initiating node first marks the *synchronization* packet with timestamp based on its own system clock (denoted by T_1) and sends it to the target node, to which it attempts to synchronize its clock. The target node records two timestamps, based on its system clock, where one is the time of receiving the *synchronization* packet (denoted by T_2) and another is the time of sending back an *acknowledgment* packet (denoted by T_3) marked with the other two timestamps, T_1 and T_2. Finally the initiating node records the timestamp of when it just receives the *acknowledgment* from the target node, denoted by T_4. Then, the propagation delay

d between two nodes, and the relative clock drift $\varDelta$ with respect to the target node are calculated as follows:

$$\begin{aligned} d &= \frac{(T_2 - T_1) + (T_4 - T_3)}{2}, \\ \varDelta &= \frac{(T_2 - T_1) - (T_4 - T_3)}{2}. \end{aligned} \tag{3}$$

In order to synchronize, the synch-initiating node increments its system clock by $\varDelta$. Propagation delay d is used to estimate the exact time of transmission of the packet from the sender, if needed.

Two Phase Synchronization

The simple method of calculating d and $\varDelta$ is sufficient in a stable network such as wired Internet, where the propagation delay d is typically identical irrespective of direction. However, in wireless sensor networks, such a nice property does not hold, motivating the synchronization based on the following two phases **P1** and **P2**.

P1. *Level discovery phase:* TPSN synchronizes the sensor nodes in a network with respect to the clock of the root node that is assigned level 0. Then, the root node starts to build a tree-like hierarchical structure by broadcasting a *level_discovery* packet to adjacent nodes. Nodes which receive the packet are assigned level 1 and continue to broadcast their *level_discovery* packets with incremented level flag. Those nodes, excluding the ones which are already assigned a certain level, are then assigned to level 2. Using this method, a tree structure, covering the entire nodes, is recursively constructed by broadcasting *level_discovery* packets repeatedly as shown in Fig. 12.

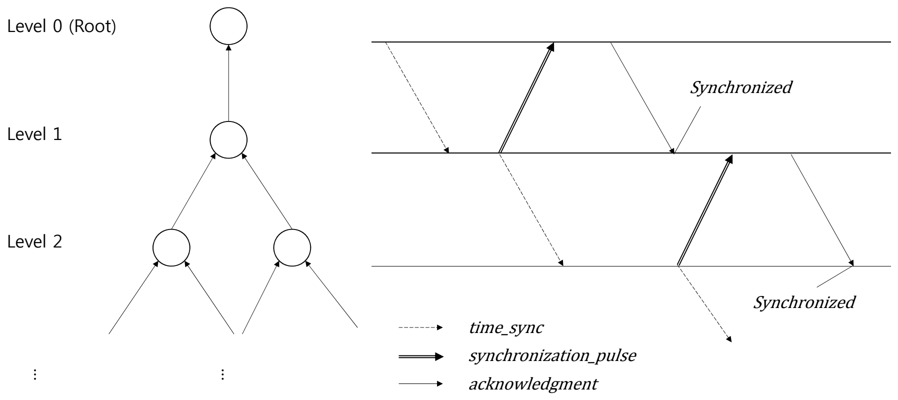

Fig. 12 Hierarchical tree structure of TPSN formed by broadcasting *level_discovery* packets.

P2. *Time synchronization phase:* After constructing the hierarchical tree, the root node initializes network-wide time synchronization by sending a *time_sync* packet to level 1 nodes. The level 1 nodes, after receiving the *time_sync* packet, respond to the root node by sending a *synchronization_pulse* marked with their timestamp, which starts the actual pairwise time synchronization. The level 1 nodes adopt random backoff before transmitting their *synchronization_pulse* in order to avoid collisions at the root node. The root node then sends back an *acknowledgement* packet to each level 1 node with its own timestamps marked. Finally, the level 1 nodes, which receive *acknowledgment* from the root node, calculate the clock drift from the packet and are synchronized to the root.

To avoid too much synchronization overhead, the *synchronization_pulse* from the level 1 nodes are recognized as a *time_sync* packet to the level 2 nodes. After receiving *synchronization_pulse* from their parent node, level 2 nodes start the pairwise time synchronization by sending their *synchronization_pulse* back to their parent with the random backoff algorithm in order for the parent node to complete their time synchronization without being interrupted. Then the level 1 nodes respond with *acknowledgment* to the level 2 nodes. This pairwise synchronization is performed throughout the network to the lowest level leaf nodes.

Comparison to RBS

TPSN adopts the hierarchical tree structure in order to decrease receiver-side complexity, being in contrast to its predecessor Reference-Broadcast Synchronization (RBS) [8]. In RBS, n receivers within a broadcast area communicate with each other to obtain time transformation table, leading its complexity to $O(n^2)$. In TPSN, the receiver-side complexity is reduced to $O(n)$ because the receiver only communicates with either of its parent node or its children nodes for time synchronization. The hierarchical structure also enables TPSN to have more scalability in the sense of high synchronization accuracy because the synchronization only depends on the parent node. Table 2 shows the results on the accuracies of TPSN and RBS, originally studied in [10].

Table 2 Accuracy Comparison: TPSN vs. RBS [10]

	TPSN	RBS
Average error (μs)	16.9	29.13
Worst case error (μs)	44	93
Best case error (μs)	0	0
Time error less than or equal to average error (%)	64	53

We comment that TPSN becomes less efficient in case when there exist too many sensors in a network or high mobility. Also, in comparison to other methods that each node contains the time transformation of adjacent nodes to convert time information, TPSN tends to be more costly because it adjusts the actual physical clock whenever the time synchronization is performed.

5 Conclusion

This chapter presents the design issues of the networking components for transferring data from sensors to sinks in an energy-efficient manner for autonomous fatigue crack detection system, which is a useful application of structural health monitoring. The key feature of our application is extremely low offered traffic load. We present and compare the strength and weakness of various design choices involving multiple layers.

Acknowledgements This work is supported by the Center for Integrated Smart Sensors funded by the Ministry of Science, ICT & Future Planning as the Global Frontier Project.

References

1. Ansari, J., Pankin, D., Mähönen, P.: Radio-triggered wake-ups with addressing capabilities for extremely low power sensor network applications. International Journal of Wireless Information Networks **16**(3), 118–130 (2009)
2. Buettner, M., Yee, G., Anderson, E., Han, R.: X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In: Proceedings of ACM SenSys (2006)
3. Cao, Y., Guo, S., He, T.: Robust multi-pipeline scheduling in low-duty-cycle wireless sensor networks. In: Proceedings of IEEE Infocom (2012)
4. Chintalapudi, K., Fu, T., Paek, J., Kothari, N., Rangwala, S., Caffrey, J., Govindan, R., Johnson, E., Masri, S.: Monitoring civil structures with a wireless sensor network. IEEE Internet Computing **10**(2), 26–34 (2006)
5. Chowdhury, K.R., Nandiraju, N., Cavalcanti, D., Agrawal, D.P.: CMAC-A multi-channel energy efficient MAC for wireless sensor networks. In: Proceedings of IEEE Wireless Communications and Networking Conference (WCNC) (2006)
6. Demirkol, I., Ersoy, C., Onur, E.: Wake-up receivers for wireless sensor networks: benefits and challenges. IEEE Transactions on Wireless Communications **16**(4), 88–96 (2009)
7. Dhanaraj, M., Manoj, B., Murthy, C.S.R.: A new energy efficient protocol for minimizing multi-hop latency in wireless sensor networks. In: Proceedings of 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom) (2005)
8. Elson, J., Girod, L., Estrin, D.: Fine-grained network time synchronization using reference broadcasts. ACM SIGOPS Operating Systems Review **36**(SI), 147–163 (2002)
9. de Francisco, R., Zhang, Y.: An interference robust multi-carrier wake-up radio. In: Proceedings of IEEE Wireless Communications and Networking Conference (WCNC) (2011)
10. Ganeriwal, S., Kumar, R., Srivastava, M.B.: Timing-sync protocol for sensor networks. In: Proceedings of ACM SenSys (2003)

11. Guo, C., Zhong, L.C., Rabaey, J.M.: Low power distributed MAC for ad hoc sensor radio networks. In: Proceedings of IEEE Global Telecommunications Conference (GLOBECOM) (2001)
12. Kim, S., Pakzad, S., Culler, D., Demmel, J., Fenves, G., Glaser, S., Turon, M.: Health monitoring of civil infrastructures using wireless sensor networks. In: Proceedings of ACM Information processing in sensor networks (IPSN) (2007)
13. Le-Huy, P., Roy, S.: Low-power wake-up radio for wireless sensor networks. Mobile Networks and Applications **15**(2), 226–236 (2010)
14. Li, Y., Ye, W., Heidemann, J.: Energy and latency control in low duty cycle MAC protocols. In: Proceedings of IEEE Wireless Communications and Networking Conference (WCNC) (2005)
15. Lu, G., Krishnamachari, B., Raghavendra, C.S.: An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks. In: Proceedings of IEEE Parallel and Distributed Processing Symposium (2004)
16. Miller, M.J., Vaidya, N.: A MAC protocol to reduce sensor network energy consumption using a wakeup radio. IEEE Transactions on Mobile Computing **4**(3), 228–242 (2005)
17. Mills, D.L.: Internet time synchronization: the network time protocol. IEEE Transactions on Communications **39**(10), 1482–1493 (1991)
18. Milosiu, H., Oehler, F., Eppel, M., Fruhsorger, D., Lensing, S., Popken, G., Thones, T.: A 3-μw 868-mhz wake-up receiver with- 83 dbm sensitivity and scalable data rate. In: Proceedings of IEEE ESSCIRC (2013)
19. Pletcher, N.M., Gambini, S., Rabaey, J.: A 52 μw wake-up receiver with -72 dbm sensitivity using an uncertain-if architecture. IEEE Journal of Solid-State Circuits **44**(1), 269–280 (2009)
20. Polastre, J., Hill, J., Culler, D.: Versatile low power media access for wireless sensor networks. In: Proceedings of ACM SenSys (2004)
21. Song, L., Hatzinakos, D.: A cross-layer architecture of wireless sensor networks for target tracking. IEEE/ACM Transactions on Networking **15**(1), 145–158 (2007)
22. Subramanian, R., Fekri, F.: Sleep scheduling and lifetime maximization in sensor networks: fundamental limits and optimal solutions. In: Proceedings of ACM Information processing in sensor networks (IPSN) (2006)
23. Sun, Y., Johnson, D.B.: RI-MAC : A Receiver-Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks. In: Proceedings of ACM SenSys (2008)
24. Texas Intruments: CC2420 2.4 GHz IEEE 802.15.4 / ZigBee-Ready RF Transceiver (Rev. C). http://www.ti.com/product/cc2420
25. Ye, W., Heidemann, J., Estrin, D.: An energy-efficient MAC protocol for wireless sensor networks. In: Proceedings of IEEE Infocom (2002)
26. Ye, W., Silva, F., Heidemann, J.: Ultra-low duty cycle MAC with scheduled channel polling. In: Proceedings of ACM SenSys (2006)
27. Yoon, D.Y., Jeong, C.J., Cartwright, J., Kang, H.Y., Han, S.K., Kim, N.S., Ha, D.S., Lee, S.G.: A new approach to low-power and low-latency wake-up receiver system for wireless sensor nodes. IEEE Journal of Solid-State Circuits **47**(10), 2405–2419 (2012)