IEEE Xplore ®
Notice to Reader

“Path Perception Synchronization Trigger Method for Dual Mode Multiplexing Mechanical Vibration Wireless Sensor Networks”

by Hao Fu, Baoping Tang, Yi Huang, and Lei Deng

published in IEEE *Access* Early Access

Digital Object Identifier: 10.1109/ACCESS.2021.3064697

It is recommended by the Editor-in-Chief of IEEE *Access* that this article will not be published in its final form.

We regret any inconvenience this may have caused.

Derek Abbott
Editor-in-Chief
IEEE *Access*

IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Path Perception Synchronization Trigger Method for Dual Mode Multiplexing Mechanical Vibration Wireless Sensor Networks

First A.Hao Fu[1],Second B.Baoping Tang[2],Third C.Yi Huang[3],Fourth D.Lei Deng[4]

[1]Hao Fu, The State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing, China
[2]Baoping Tang,The State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing, China
[3]Yi Huang,The State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing, China
[4]Lei Deng，The State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing, China

First A. Hao Fu (e-mail: fuhowe@qq.com)

This work was supported in part by the This work was spported by the National Natural Science Foundation of China under Grant 51675067 and 51775065

Abstract Synchronization triggers for mechanical-vibration wireless sensor networks play an important role in data acquisition and the analysis of multiple nodes. At present, most nodes are designed using a single protocol, which makes it difficult to achieve a balance between network establishment and synchronization. Meanwhile, such synchronization trigger methods rely on a clock, beacon, or communication between nodes. A method based on a clock and beacon requires frequent timestamps, which reduces the synchronization trigger accuracy and increases the power consumption. The communication between nodes requires these nodes to be at a close distance; otherwise, the node cannot receive synchronization information. Aiming at the above problems, a path perception synchronization trigger method based on dual-mode multiplexing and cluster networks is proposed. First, the node is designed using dual mode multiplexing. Second, an adjoint networking method is proposed to construct the cluster network topology. Next, a path perception time measurement method based on the terminal device and terminal assistant measurement without clock synchronization, beacon synchronization, and communication between nodes is discussed. Finally, a maximum time deviation compensation and time block synchronization trigger measurement method are proposed to solve the problems of a time delay compensation and system instruction interruption. The experimental results show that the maximum time error of the path perception synchronization trigger is between 50 and 60 μs, and the minimum time error is between 10 and 16 μs, which significantly improves the synchronization triggering accuracy under conditions that do not rely on clock synchronization, beacon synchronization, or communicating between nodes.

INDEX TERMS Mechanical equipment monitoring, dual modes multiplexing, path perception, wireless sensor networks, synchronization trigger

I. INTRODUCTION

Intelligent equipment is an essential symbol of the modern manufacturing industry. It integrates the smart operation and maintenance system by taking advantage of information technology [1–3]. Hence, the monitoring approach of the equipment condition achieves a significant improvement. Wireless sensor networks (WSNs) are a promising multidisciplinary technology based on the integration of sensors and information technology to monitor physical or environmental conditions. WSNs are easy to deploy on monitoring targets with the advantages of a simple harness and a convenient acquisition. In addition, the sensor node can be deployed on a sealed space that a wired system is unable to set up. WSNs have recently been widely applied to monitor conditions such as coal mines [4], mechanical equipment [5], bridge condition monitoring [6], and wind power generation [7].

Despite the apparent successful application without cable and ease deployment, researchers still have concerns and issues regarding the use of WSNs for mechanical equipment monitoring. In general, a vibration signal is a distinct feature of rotating mechanical equipment, and the requirement of clock synchronization for a vibration signal is extremely high in most cases [8, 9]. However because the sensor node disperses spatial discrete positions, the time and spatial jittering during synchronous acquisition between nodes will affect the validity of the signal acquisition and the result of the state analysis. Thus, researchers have focused on synchronous trigger methods.

Considering the high sample frequency and high sample accuracy of vibration data, most studies have adopted a data processing core cooperating with a single transmission core. Bengherbia, et al. [10] used a field-programmable gate array (FPGA) processor as the control core and a Nordic Radio Frequency (NRF) wireless communication processor as the wireless transceiver and receiver, of which the highest data transmission rate reaches 2 Mbps, and the analog-to-digital converter (ADC) is 16 bits. In a previous study [11], Huang developed a sensor node with a sample rate and sample accuracy of up to 20 kHz and 16 bits, respectfully, which is equipped with a control core STM32 and a transmission core CC2530. Among these sensor nodes, a single-hop network, namely, a star network with synchronization trigger accuracy, has been realized. In addition, Xiao et al. [12] proposed a high-accuracy synchronous acquisition algorithm with multi-hop sensor networks using the start frame delimiter (SFD) signal of a broadcast beacon frame. However, the 250-kbps data rate limits the transmission performance of the network, particularly when all sensor nodes deliver data to the monitoring center.

Based on the developed sensor node, the existing synchronous algorithms can be classified into three categories: beacon-based (e.g., RBS [13], TPSN [14]), timestamp-based (e.g., FTSP [15], DMTS [16]), and communication between acquisition nodes [17]. Moreover, Xiao [12] proposed a multiple-hop cumulative error control method based on cross-layer synchronization acquisition. By establishing a key information transmission model of synchronization acquisition, the cumulative mechanism of a synchronization error in multi-hop networks was analyzed, and the results of the cumulative synchronization error caused by the scheduling algorithm of the embedded operating system and crystal oscillator offset on the hardware were obtained. The nodes then delay the timeslot to simultaneously trigger an acquisition. Wang et al. [18] improved the Flooding Time Synchronization Protocol (FTSP) algorithm by obtaining a timestamp. A re-sampling was then used after data acquisition for data processing to improve the synchronization trigger and synchronous acquisition accuracy.

Based on the above discussion, the design of the sensor node for monitoring the mechanical equipment is expected to overcome several challenges such as a high synchronous accuracy, optimal network bandwidth allocation, and low energy consumption. Moreover, the existing synchronous acquisition algorithms rely on a beacon or timestamp, given the monitoring of harsh scenarios, which may result in a failure of network communication between nodes over a long distance. In addition, the sensor nodes equipped with a single controller cooperating with a transmitter makes it difficult to balance the transmission rate, distance, and synchronization accuracy.

In summary, to solve the aforementioned problems, a path perception synchronization trigger method without clock synchronization, beacon synchronization, and communication between nodes is proposed based on dual-mode multiplexing. The key contributions and highlights of this research are outlined below.

(1) To balance the demands of the transmission speed, transmission distance, networking establishment, and synchronization trigger, the sensor node is designed using a dual mode, and adjoint networks establish a method for the mechanical vibration of wireless sensor networks.

(2) To solve the problem of relying on beacon synchronization, clock synchronization, and communication between nodes, a path perception synchronization trigger method for mechanical vibration wireless sensor networks is proposed.

The remainder of this paper is organized as follows. The cause of a synchronization triggering cumulative error analyses in a cluster network, the method of network establishment based on dual modes, and a data transmitting protocol frame design are presented in Section II. Section III describes the path perception synchronization trigger method, including the path time measurement, maximum difference in path time compensation, and time block processing of the delay time. The experiments conducted to confirm the appropriate path perception times and compare the synchronization trigger errors are described in Section IV. Finally, Section V provides some concluding remarks regarding this paper.

II. CLUSTER NETWORKS AND CUMULATIVE ERROR

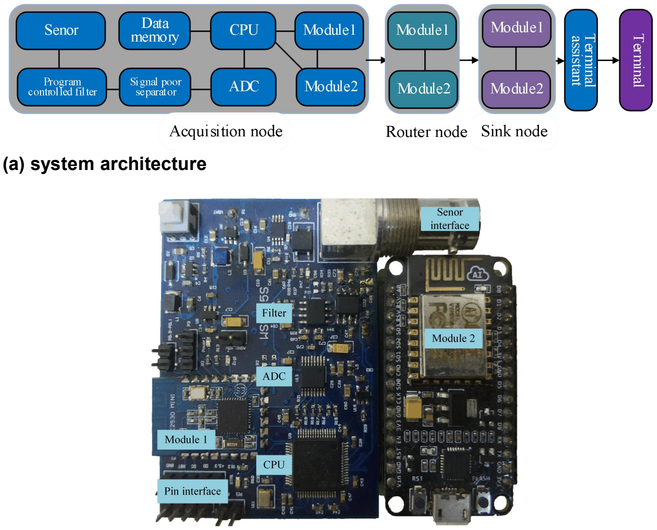
In this section, the adjoint network establishment based on dual modes and a data protocol frame design are introduced. In addition, the cluster network synchronization trigger error is analyzed.

A. Node Design and Network Establishment Method

1) DUAL MODE MULTIPLEXING NODES

In general, the front-end data processing architecture of the acquisition node remains unchanged, which is designed using an ARM processor STM32 as its data processing core with a 24-bit ADC (e.g., AD7766). Differing from previous studies [11, 12] the transmission module adopts dual modes, i.e., two radio frequency (RF) modules. The RF modules must accomplish a number of goals, i.e., a high transmission rate for burst vibration data, a self-organization network, and a high synchronization accuracy. Therefore, module 1 based on IEEE 802.15.4 (e.g., CC2530) is adopted to establish the main networks and transmitting address information. The IEEE 802.11-based module 2 (e.g., ESP8266) is taken as the transmitting vibration data and synchronization trigger. The acquisition front-end is multiplexed by the two aforementioned RF modules to balance the requirements of transmission, networking, and synchronization.

Inside the acquisition node, data are sent through a wired transmission using a universal synchronous/asynchronous receiver/transmitter (USART), serial peripheral interface (SPI), and secure digital input and output (SDIO). By contrast, the data are communicated by the two RF modules with wireless connections. When a command is issued from the terminal device or terminal assistant to the acquisition node, the acquisition node



(a) system architecture

(b) node porotype

FIGURE 1. The architecture of the dual modes multiplexing node and its porotype

can directly receive the command, or the command is forwarded through single or multiple routers on the way to the acquisition node. The central processor unit (CPU) responds immediately after receiving the command through USART. The architecture of the dual-mode multiplexing node is shown in Fig. 1(a) and the acquisition node porotype is shown in Fig. 1(b).

2) ADJOINT NETWORKS

Considered key units in terms of data transmission for mechanical equipment monitoring, the RF chips selected must meet certain trends such as a high throughput, self-organization, and high synchronous trigger accuracy. The current single RF-based module of the node only considers one aspect, for example, a CC2530 or an ESP8266. To take full advantage of these two RF modules, a sensor node is equipped with both a CC2530 and an ESP8266. As mentioned above, the CC2530 is responsible for network construction and the ESP8266 takes responsibility for the data transmission. Therefore, before the node transmits the data, the node should know the network relationship of the ESP8266 in advance. To solve these problems, a new dual mode adjoint network method is proposed. The dual-mode network method consists of main and adjoint networks. A topology information exchange can be realized between the two protocols in a wired approach. Fig. 1 shows a schematic of the short addresses of the main network and the dual mode adjoint networks.

3) MAIN NETWORKING NODE ADDRESSES

To establish adjoint networks the main network addresses are obtained, and the main network address information can then be sent to the adjoint nodes.

The number of address C_d that can be assigned to the router subset from the parent devices is calculated in (1).

$$C_d = \begin{cases} 1+C_m(L_m - d - 1), R_m = 1 \\ \dfrac{1+C_m - R_m - C_m \times R_m^{L_m - d - 1}}{1 - R_m}, R_m \neq 1 \end{cases} \tag{1}$$

where C_m is the maximum number of child nodes that the parent device can associate with, R_m is the maximum number of routers as its children, and L_m is the maximum network depth. In addition, C_d indicates that the address numbers of a child device with router capacity can be allocated under the condition of given network depth d.

If the value of C_d is zero, it indicates that the node has no ability to associate with the child node. If the network depth is d, the network addresses of the acquisition node n and router node n are as demonstrated in (2)[19]:

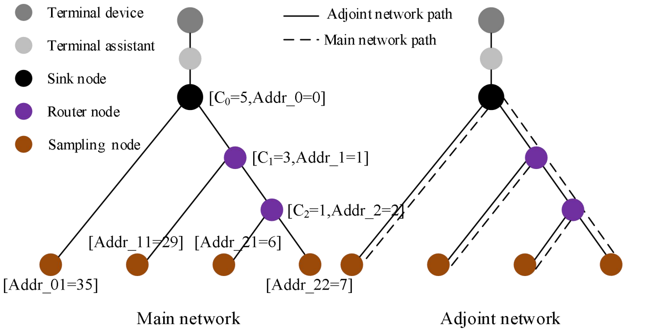


FIGURE 2. Dual modes adjoint networks address and structure

$$\mathrm{A_n} = \begin{cases} A_p + C_d \times n_r + 1, & \text{router node} \\ A_p + C_d \times R_m + n, & \text{acquisition node} \end{cases} \tag{2}$$

where A_n is the current node address, A_p is the address of the parent node of the current node, n_r is the number of child routers that have been associated with the parent node, and n is the serial number of the current node.

Specifying $R_m = 3$, $C_m = 8$, and $L_m = 3$, the constructed networks are shown in Fig. 2. Therefore, the value of C_d can be computed using (1), as shown in (3)–(6).

$$C_0 = 1+C_m(L_m - d - 1) = 1+8*(3-0-1) = 17 \tag{3}$$

$$C_1 = 1+C_m(L_m - d - 1) = 1+8*(3-1-1) = 9 \tag{4}$$

$$C_2 = 1+C_m(L_m - d - 1) = 1+8*(3-2-1) = 1 \tag{5}$$

$$C_3 = 1+C_m(L_m - d - 1) = 1+8*(3-3-1) = -7 \tag{6}$$

When d is equal to 3, node address cannot be allocated. Thus, $C_3 = 0$. To obtain the short address of each acquisition and router node, (2) and the short addresses below, i.e., (7)–(13) are computed.

$$Addr\_0 = 0 \tag{7}$$

$$Addr\_1 = Addr\_0 + C_0 \cdot 0 + 1 = 0 + 17*0 + 1 = 1 \tag{8}$$

$$Addr\_2 = Addr\_1 + C_1 \cdot 0 + 1 = 1 + 17*0 + 1 = 2 \tag{9}$$

$$Addr\_01 = Addr\_0 + C_0 \cdot R_m + n = 0 + 17*2 + 1 = 35 \tag{10}$$

$$Addr\_11 = Addr\_1 + C_1 \cdot R_m + n = 1 + 9*3 + 1 = 29 \tag{11}$$

$$Addr\_21 = Addr\_2 + C_2 \cdot R_m + n = 2 + 1*3 + 1 = 6 \tag{12}$$

$$Addr\_22 = Addr\_2 + C_2 \cdot R_m + n = 2 + 1*3 + 2 = 7 \tag{13}$$

4) NETWORKS PATH COST

The path cost is an important index for finding the most effective path during the process of establishing the main networks. During router discovery, routing algorithms use metrics to compare the routers. To calculate this metric, the routing protocol defines the link cost as the link measure of each path, and the path cost is defined as the sum of all link costs on the path. The calculation formula for the path cost is shown in (14).

$$C\{P\} = \sum_{i=1}^{L\text{-}1} C[D_i, D_{i+1}] \tag{14}$$

where the path length is L and the path is P. The sequence of the path is $[D_1, D_2, \ldots\ldots D_L]$. In addition, $[D_1, D_2]$ is the sub-path with length 2, and $C\{[d_i, d_{i+1}]\}$ is the link cost. The link cost $C\{1\}$ of link 1 is a function in [0,......,7], which is defined as follows:

$$C\{j\} = \begin{cases} 7 \\ \min\left(7, round(\frac{1}{p_i^4})\right) \end{cases} \tag{15}$$

where P_i is the probability that data will be transmitted on link i, and round(x) is the integer value that returns the rounding of x. There are two methods used to obtain the value of P_i. One method is to evaluate the actual times of receiving the beacon and data frame within a fixed time range; the other method is to calculate the average value of the link quality indicator of each frame provided by the media access control and the port physical layer.

5) STEPS OF ESTABLISHING ADJOINT NETWORKS

There are two steps in the construction of dual mode adjoint networking: establishing the main networks and establishing adjoint networks. Module 1 only participates in the construction of the main networks, and the main network is organized normally. After finishing the construction of the main networks, the topology of the main network is analyzed at the terminal device, and the adjoint network topology information including the service set identifier (SSID) and Wi-Fi Protected Access (WPA) with module 2 is generated. The topology information of the main networks can be obtained using (7)–(13) [17]. The adjoint networks are constructed under the condition that the main networks are completed, which is divided into two steps. First, the topology structure generated by the main networks is obtained, and the topology information is sent to the terminal device of the main network for analysis and storage. The topology information of the SSID and WPA corresponding to module 1 is sent to module 2 through USART. Immediately, SSID and the WPA of module 2 are analyzed and set, after which the nodes at all levels start to wirelessly connect their parent SSID. Thus, the sink node does not need to connect to the parent node, and the acquisition nodes do not need to set their own SSID.

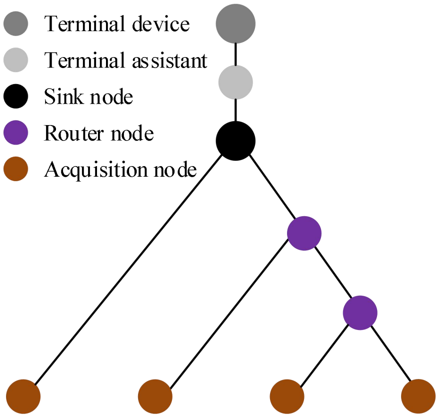


FIGURE 3. Typical cluster networks topology

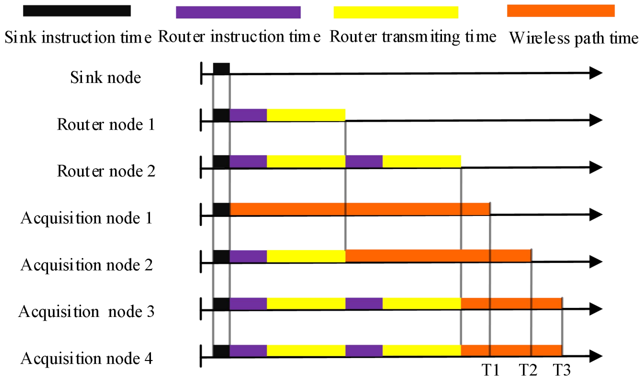


FIGURE 4. Synchronization sequence of cluster network

B. CUMULATIVE CLOCK ERRORS OF CLUSTER NETWORKS

Cluster networks are a commonly used network topology in mechanical vibration monitoring owing to the simple route path. The synchronization acquisition error mainly occurs at the beginning of the acquisition. Because the nodes are in a heterogeneous spatial network structure, the acquisition command arrives at the acquisition node at different times. In addition, ambient temperature, humidity, and electromagnetic interference lead to jitters in time and space [12] during the acquisition process. However, this is influenced less by a practical acquisition owing to the short acquisition time and relatively stable ambient environment. Fig. 3 shows a typical cluster network topology.

During the process of data acquisition, the main cause of the command arriving at each acquisition node at different times is the diversity among the transmission path times, which include the data transmission and program instruction execution times. The transmission path time refers to the time consumed during a wireless transmission where the command travels from the terminal device or terminal assistant to the acquisition node. The data transmission time refers to the time when the data are transmitted between the sink node, router nodes, and acquisition nodes. The program instruction execution time refers to the time when all program instructions are executed for each node.

Fig. 4 shows the time sequence of the cluster networks without applying a synchronization method. The command is emitted from the sink node. Theoretically, the shorter the path is and the smaller the number of intermediate routers, the shorter the time required for the acquisition node to receive the command. Acquisition node 1 receives the trigger command at T1 earlier than acquisition node 2. Acquisition node 2 receives the command at T2 compared to nodes 3 and 4. However, acquisition nodes 3 and 4 theoretically receive the trigger command at the same time as T4, and small errors occur in practical application owing to the sequence of the instruction execution.

C. DATA FRAME PROTOCOL

A reasonable data protocol frame can not only ensure that the command is accurately sent to the corresponding acquisition node but can also be treated as the basis for the synchronization trigger.

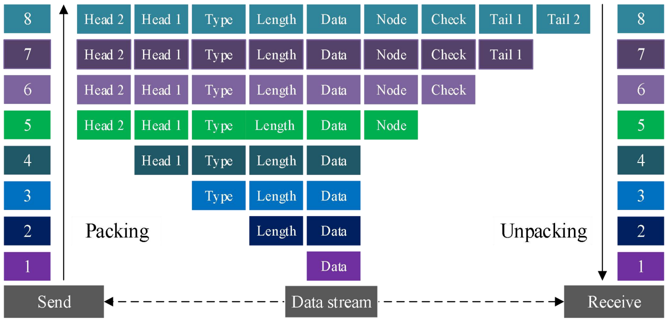

8 Head 2 Head 1 Type Length Data Node Check Tail 1 Tail 2 8
7 Head 2 Head 1 Type Length Data Node Check Tail 1 7
6 Head 2 Head 1 Type Length Data Node Check 6
5 Head 2 Head 1 Type Length Data Node 5
4 Head 1 Type Length Data 4
3 Type Length Data 3
2 Packing Length Data Unpacking 2
1 Data 1
Send
Data stream
Receive

FIGURE 5. **Packing and unpacking of protocol data frame**

In this section, the protocol frame design, protocol packaging, and unpacking design are described.

Table 1 shows the basic format of the acquisition command protocol frame. Transmission protocols are divided into downstream and upstream protocols. The downstream protocol is a format by which the data are sent from the terminal to the acquisition node, and the upstream protocol is a format for reverse transmission of the data. A complete multiple parameter configuration protocol consists of seven parts: frame head, data type, data length, data block content, acquisition node number, check code, and frame tail. The frame header has 2 bytes, including 1 byte of the data type and 1 byte of the data length. There are three situations for the data content, where 14 bytes are for configuring the parameters, 1 byte is for issuing control commands, and 120 bytes are for uploading the vibration data.

TABLE 1
ACQUISITION COMMAND PROTOCOL DATA FRAME

Data frame	Number	Content	Introduce
Frame head 1	1	0xAA	Fixed
Frame head 2	2	0xDD	Fixed
Data Type	3	0x2	Fixed
Data Length	4	0x01	Fixed
Data Content	5	0x01	Variable
Node serial	6	0x00	Variable
Check code	7	0xFF	Fixed
Frame tail 1	8	0xDD	Fixed
Frame tail 2	9	0xAA	Fixed

In addition, the node number is 1 byte, the check code is 1 byte, and the frame end is 2 bytes. In summary, the total data length of a transmission protocol is 9, 22, or 128 bytes.

The data are packed according to the corresponding protocol when transmitted in the upstream and downstream regions and unpacked when the data are transmitted and received. The process is shown in Fig. 5. When the data are packed, the process starts with the data content at the center and moves toward both ends, during which the data length is added before the data are packed. The data length is the number of bytes of the data content. The data type is then forwarded. Next, frame heads 2 and 1 are forwarded in order. After the front part of the protocol packing is completed, the latter part starts behind the data content. The node number, the fixed check code, and frame tails 1 and 2 are added behind the data content.

When the protocol is unpacked, it proceeds from both ends until reaching the central data content. First, whether the data contents of the front and end of the data are both 0xAA is determined. If so, the unpacking starts and the data contents of the second-front and second-end are checked using 0xDD. The next unpacked object is the check code. If the check code is correct, it starts to unpack the node number. The command type is then unpacked in the case of a correct check code. Finally, the command type and command length are unpacked. In the above parsed steps, if any data errors occur during any step, the parsing is finished and the data protocol error is fed back up or down. Through the packing and unpacking of the above protocol, the reliability of the data can be ensured through wired and wireless transmission.

III. PATH PERCEPTION SYNCHRONIZATION TRIGGER METHOD

In this section, synchronization trigger methods, with a terminal assistant and without a terminal assistant, are proposed. A terminal assistant is applied as the final scheme to analyze the path perception synchronization trigger. These symbols are used in the formulas shown in Table 2.

TABLE 2
SYMBOLS LIST

Symbol	Indication	Symbol	Indication
$\varDelta_{tij}$	Time of minimum increment	$Max(\overline{T}_{wij})$	maximum time of single path
F_i	frequency of CPU	$\varDelta_i$	compensation time of node *i* single path
i	serial number of Node	T_{ci}	time of single instruction
j	measurement count of Node *i*	n	number of state cycle
t_{sij}	start time of sending command	q	number of clock cycle
t_{rij}	end time of receiving command	T_m	machine cycle
m	the *m-th* times measurement of node	T_s	state cycle
T_{wij}	time of single path	T_t	clock cycle
$\overline{T}_{wij}$	average Time of single path	N	number of Single instruction execution
t_{cij}	instruction time of Node	M	number of time packet
t_{tij}	transmit time of Router	K	value of N times M

A. PATH PERCEPTION METHOD

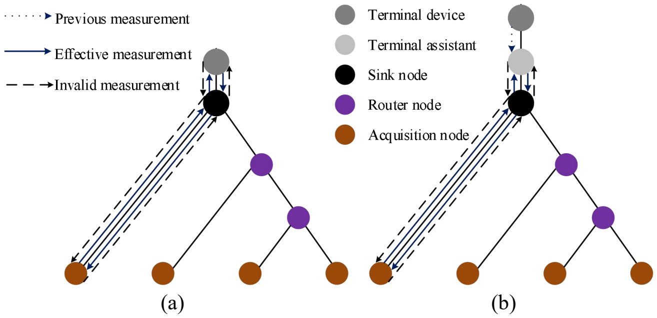


FIGURE 6. **Timing measurement with acquisition node**

The function of a path perception time measurement is to obtain the transmission time required for commands on different paths. In this paper, a path perception time measurement method is proposed based on the acquisition node timing, terminal device timing, and terminal assistant timing. Fig. 6(a) shows the timing measurement with an acquisition node without a terminal assistant. Fig. 6(b) shows the timing measurement with the acquisition node including the terminal assistant.

1) PATH TIME MEASUREMENT WITH ACQUISITION NODE TIMING

In the case of a clock measurement with an acquisition node without a terminal assistant, the timing core is the acquisition node. The terminal device sends the first measurement command. The acquisition node starts the timing at the arrival of the first measurement command. After the timing command is executed, the second measurement command is sent to the terminal device. The terminal device immediately sends the third measurement command to the acquisition node as feedback. The acquisition node immediately stops the timing after receiving the third measurement command. The difference in time between the second and third measurements is calculated, which will be sent to the terminal device and applied to the unified processing. Among them, the second and third measurements are effective, between which multiple program instructions should be processed by a computer with an operating system. The number of instructions is proportional to that of the nodes. When multiple instructions are executed, the polling of the terminal operating system will cause the instruction to be interrupted in milliseconds. This results in a time discontinuity of the measurement command sent, which reduces the synchronization accuracy.

In the case of a timing measurement with an acquisition node including a terminal assistant, except for the terminal device, which needs to initiate a premeasurement command, the other steps are the same as above. There is no operating system used during the process of sending formal commands, and thus the problem of polling interrupt instructions can be solved. However, this method uses the processor of the acquisition node as the timing reference. Thus, the frequency is generally low, resulting in a large minimum time increment, which makes it difficult to meet the high precision synchronization requirements.

2) PATH TIME MEASUREMENT WITH TERMINAL DEVICE AND TERMINAL ASSISTANT TIMING

The time core is the terminal device, as shown in Fig. 7(a), or the terminal assistant, as shown in Fig. 7(b). During the process of terminal device measurement, the timing and measurement commands are controlled by the terminal devices with the same measurement steps as the second and third steps mentioned above. Because there are still multiple program instructions for the measurement command, there is also a problem in that the polling of the operating system causes the instruction to be interrupted within milliseconds. For a measurement with a terminal assistant, the start measurement commands are controlled by the terminal device. Once the premeasurement command is sent by the terminal device, the control core is delivered to the terminal assistant and the remaining steps are the same. The terminal assistant does not need any operating system intervention during the measurement, which avoids the problem of an instruction interruption. A super high-frequency processor can be selected to reduce the minimum time increment. A single measurement is also an effective measurement, and can reduce the measurement time. Therefore, a terminal assistant measurement is utilized in this study.

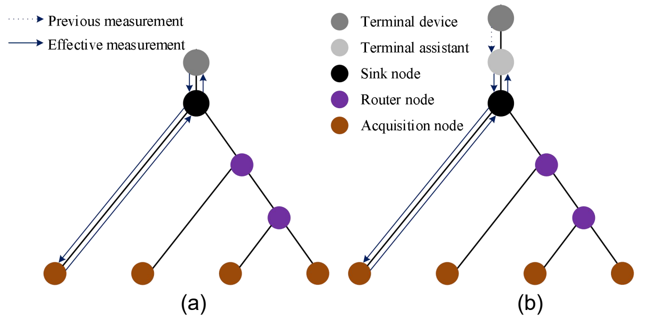


FIGURE 7. **Timing measurement with terminal device and termina assistant**

3) MEASUREMENT STEPS WITH TERMINAL ASSISTANT

Once a path perception is required for each router during the terminal assistant measurement, because the terminal assistant cannot confirm when to initiate the measurement, a premeasurement command is issued to the terminal device before the formal measurement. Control privilege is then committed to the terminal assistant. Before the measurement, a software timing assistor is assigned to each acquisition node. Formal measurement commands are then sent to each node. Meanwhile, the time starts immediately. A feedback command is sent by each acquisition node at the arrival of the measurement command. The terminal assistant stops the timing after receiving the node feedback commands. The time difference is calculated and sent to the terminal device for data processing.

The accuracy of the terminal assistant measurement mainly depends on the clock frequency. The calculation of the minimum time increment Δ_{ti} is shown in (16), where F_i is the reference frequency of the terminal assistant. The smaller the clock frequency, the lower the time accuracy measured by the path perception and the lower the accuracy of the synchronization trigger.

$$\Delta_{ti} = \frac{1}{F_i} \tag{16}$$

The terminal assistant timing measurement can compensate for the shortcomings of the above method and has a super-high synchronization accuracy. A terminal assistant timing measurement is used to analyze the measurement method in this study. Each node is numbered i before the measurement. The terminal device sends a measurement command to node i and starts the timing immediately. The initial time is recorded as t_{sij}. The acquisition node i immediately returns the measurement command to the terminal device after receiving the measurement command. While the terminal device receives the measurement command, the timing stops immediately, and the time of receiving the measurement command is recorded as t_{rij}. The above steps are for the one-path time measurement of a single node. The results of a large number of repeated measurement experiments show that the time difference of each measurement has a small error owing to the influence of time and space jitters. To reduce such errors, a method for averaging the values of the repeated measurements is adopted. The number of measurements at each node is *m*. To avoid channel competition, the measurements of the next node are not conducted until that of the previous node is completed. The subscript *s* represents sending a formal measurement command, *r* represents receiving feedback, *i* is the *i-th* node, and *j* is the *j-th* measurement.

B. COMPENSATION METHOD OF MAXIMUM TIME DIFFERENCE

The calculation of the time compensation is conducted through the recorded data after *m* tests of all nodes are completed. Here, T_{wij} is a one-way time of the *j-th* path perception measurement of the *i-th* acquisition node and is calculated in (17). The average time of $\overline{T}_{wij}$ of *m* tests is shown in (18), where t_{cij} is the command execution time of the acquisition node, terminal device, and terminal assistant, and t_{tij} is the transmit time of the data on the router node.

$$T_{wij} = \frac{1}{2} \cdot \left(t_{rij} - t_{sij} + t_{cij} + t_{tij} \right) \tag{17}$$

$$\overline{T}_{wij} = \frac{1}{m} \cdot \sum_{j=1}^{m} T_{wij} \tag{18}$$

The following describes the method for calculating the maximum path time. First, the memory is set, the initial value of which is the average time of the first acquisition node. Then, the average time of each node is compared with the memory value. The value of the memory is then exchanged if the average time of the next node is greater. The above steps are then repeated. The number of comparisons is the number of nodes. The memory value after the last comparison is the maximum path time $Max(\overline{T}_{wij})$. The one-way average path time compensation value Δ_i of each node can be obtained by subtracting the average time of each node from the maximum value. This is shown in (19).

$$\Delta_i = Max(\overline{T}_{wij}) - \overline{T}_{wij} \tag{19}$$

To obtain (20), (18) is substituted into (19), as shown below:

$$\Delta_i = \frac{1}{m} \cdot \left[Max(\sum_{j=1}^{m} \overline{T}_{wij}) - \sum_{j=1}^{m} \overline{T}_{wij} \right] \tag{20}$$

To obtain (21), (17) is substituted into (20), as shown below:

$$\Delta_i = \frac{1}{2m} \cdot \left[Max(\sum_{j=1}^{m} (t_{rij} - t_{sij} + t_{cij} + t_{tij})) - \sum_{j=1}^{m} (t_{rij} - t_{sij} + t_{cij} + t_{tij}) \right] \tag{21}$$

Because the instruction execution time t_{cij} is the same as the transmit time t_{tij} during each measurement, the compensation time, as shown in (22) of each node is related only to the test times and the difference in the path time.

$$\Delta_i = \frac{1}{2m} \cdot \left[Max(\sum_{j=1}^{m} (t_{rij} - t_{sij})) - \sum_{j=1}^{m} (t_{rij} - t_{sij}) \right] \tag{22}$$

It can be seen from (22) that the time compensation is constant after the path perception time measurement of each node.

C. TIME BLOCK PROCESSING METHOD

The path time that should be compensated for each acquisition node can be calculated using (22). The acquisition command of each node can be sent after a delay in the corresponding compensation time. Because the current terminal assistant generally uses a timer for the timing, the timer frequency is obtained through multiple frequency divisions. Thus, directly using the timer frequency for timing will result in a large minimum time increment, which makes it difficult to meet the accuracy requirements. To solve the above problems, the main frequency of the terminal assistant is set as the minimum time increment. The time delay is realized by running a single instruction. The single instruction time T_c is calculated using (23). A single instruction cycle consists of one machine cycle, *n* state cycles, or *q* clock cycles, where *n* and *q* are related to the type of terminal assistant. When the terminal assistant is fixed, *n* and *q* are fixed values.

$$T_{cij} = T_{mj} = n \cdot T_{sij} = q \cdot T_{tij} = q \cdot \Delta_{tij} = \frac{q}{F_i} \tag{23}$$

It can be seen from (23) that when the main frequency of the terminal assistant is greater, the single instruction execution time is shorter, the delay time increment is smaller, and the synchronization trigger accuracy is higher. Moreover, a long running time of a single instruction will cause abnormal conditions for the terminal assistant, and thus it is necessary to solve the problem of a long time delay.

The reference frequency of the terminal assistant was obtained before the formal measurement. The calculation for the execution time of each instruction is shown in (23). The instruction time can generally reach the micro-nanosecond level. Therefore, the number of times *N* that the single instruction time needs to be executed can be obtained by dividing the single instruction time by the compensation time. The calculation is shown in (24).

$$N_i = \frac{\Delta_i}{T_{ci}} \tag{24}$$

When *N* is large, the single instruction program occupying the processor operating system for a long period of time will trigger a "watchdog," and a large running memory will cause the system stack memory to overflow, leading to system failures and an imprecise delay. Therefore, the encapsulation process is conducted based on the single instruction time. This means that the number of single instruction executions *N* is divided into *M* time periods. The total number of executions is *K*. The stack memory of the CPU is cleaned up after *K* single instructions are executed, which can solve the stack memory problem of the terminal assistant. The relationships among *N*, *M*, and *K,* are shown in (25).

$$N_i = M_i \cdot K_i \tag{25}$$

To obtain (26), (24) and (25) are merged, as shown below:

$$\Delta_i = M_i \cdot K_i \cdot T_{ci} \tag{26}$$

To obtain (27), (23) is substituted into (26), as shown below:

$$\Delta_i = M_i \cdot K_i \cdot T_{ci} = M_i \cdot K_i \cdot \frac{q}{F_i} \cdot \tag{27}$$

To obtain (28), the format of (27) is changed into (28), as shown below:

$$N_i = M_i \cdot K_i = \frac{F_i}{q} \cdot \Delta_i \tag{28}$$

To obtain (29), (22) is substituted into (28), as shown below:

$$N_i = \frac{F_i}{q} \cdot \frac{1}{2m} \cdot \left[Max(\sum_{j=1}^{m} (t_{rij} - t_{sij}) - \sum_{j=1}^{m} (t_{rij} - t_{sij}) \right] \tag{29}$$

To reduce the jitter in time and space, the CPU frequency of the terminal assistant must be acquired before starting. Thus, the frequency matrix of each acquisition node for a path perception is presented in (30) as follows:

$$F_i = f_{(i \times 1)} = \left[f_1 \quad f_2 \quad f_3 \quad f_4 \quad \cdots \quad f_i \right]^T \tag{30}$$

To obtain (31), (30) is substituted into (29), as shown below:

$$N_i = \frac{1}{2mq} \cdot \left[Max(\sum_{j=1}^{m} (t_{rij} - t_{sij}) - \sum_{j=1}^{m} (t_{rij} - t_{sij}) \right] \times F_i \tag{31}$$

To obtain (32), (31) is merged as follows:

$$N_i = \frac{1}{2mq} \cdot \begin{bmatrix} f_1 \cdot \left(Max(\sum_{j=1}^{m} (t_{r1j} - t_{s1j}) - \sum_{j=1}^{m} (t_{r1j} - t_{s1j}) \right) \\ f_2 \cdot \left(Max(\sum_{j=1}^{m} (t_{r2j} - t_{s2j}) - \sum_{j=1}^{m} (t_{r2j} - t_{s2j}) \right) \\ \vdots \\ f_i \cdot \left(Max(\sum_{j=1}^{m} (t_{rij} - t_{sij}) - \sum_{j=1}^{m} (t_{r1j} - t_{sij}) \right) \end{bmatrix} \tag{32}$$

From (22), each node executes the number of single instructions relative to the CPU frequency, which is obtained at the beginning of the acquisition data. This means that the numbers of single instruction executions M and K used by each node for a delay are fixed. When the encapsulated time period M is large and K is small, the stack resources of the terminal assistant will be heavily occupied as a result. In addition, when M is small and *K* is large, the stack memory will be frequently cleared, which occupies the processor time. Therefore, in practical applications, it is necessary to balance the values of *M* and *K* according to the different reference frequencies of the terminal assistant. Through the above-mentioned delay time packaging method, the problem of a low accuracy of ordinary timers can be effectively solved.

When starting the acquisition, *m* path perceptions of each node are conducted based on the construction of the network. The path time of each node is then obtained. After the measurement of all nodes is completed, the compensation time of each node is obtained through an analysis and processing by the terminal assistant. Then, the synchronization trigger command is sent sequentially after a delay according to the delay compensation value of each node. Because the path time of each node is different, the moment when the acquisition command finally arrives at each acquisition node is the same after the time compensation delay. The acquisition can be triggered synchronously within the error range.

IV. EXPERIMENTS AND DISCUSSION

In this section, experiments on the path perception times and trigger error are described, and the performance of the path perception synchronization measurement is shown.

A. *Experiment of Path Perception Time*

To reduce the measurement error, an averaging of the path perception time of multiple measurement methods is adopted. To determine the number of advance measurements *m*, multiple measurements were conducted on each node. The average path time and compensation time for each node are shown in Table 3 and Fig. 8.

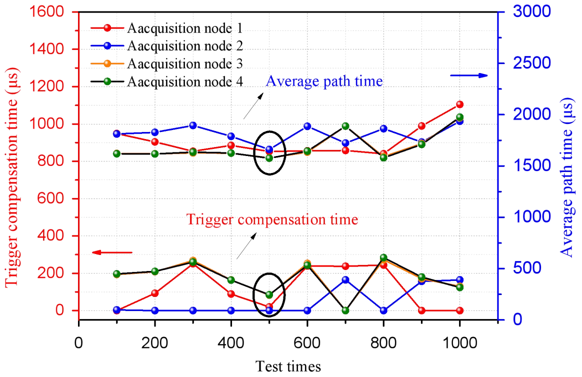


FIGURE 8. Timing measurement with terminal device and termina

It was demonstrated that the average path time and the compensation time of the synchronization trigger are generally stable with an increase in the test times. Furthermore, there is a rapid increase under 800 to 1000 tests. The average path time and

TABLE 3
TIMES OF PATH SENSING MEASUREMENTS

Option	Average single path time /μs				Trigger compensation time/μs			
Times	Node 1	Node 2	Node 3	Node4	Node1	Node 2	Node 3	Node 4
100	1815.38	1811.68	1621.95	1618.7	0	3.707	193.43	196.627
200	1734.25	1827.31	1619.04	1617.4	93.06	0	208.26	209.88
300	1643.77	1894.68	1626.20	1634.5	250.91	0	268.47	260.12
400	1700.11	1788.73	1624.62	1625.5	88.621	0	164.11	163.14
500	1641.03	1660.81	1578.34	1575.6	19.773	0	82.468	85.13
600	1647.56	1886.64	1633.81	1643.9	239.07	0	252.83	242.73
700	1650.84	1723.53	1887.81	1888.2	237.37	164.68	0.40	0
800	1618.97	1863.26	1590.72	1579.5	244.28	0	272.53	283.72
900	1889.60	1732.78	1718.56	1709.8	0	156.82	171.03	179.78
1000	2099.66	1935.36	1969.83	1975.8	0	164.30	129.83	123.86

compensation time of all nodes are at a low value when the number of basic tests is approximately 500. Therefore, it is more appropriate to measure the path perception time by approximately 500 times.

B. *Experiment of Synchronization Trigger Accuracy*

1) EXPERIMENTAL SETUP AND METHOD

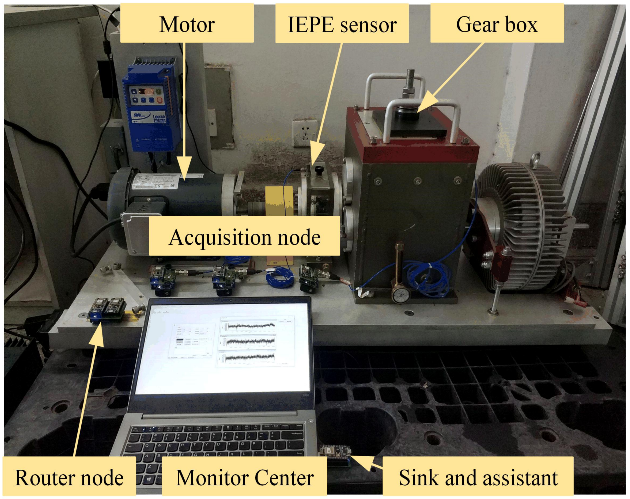


FIGURE 9. Synchronization trigger accuracy test environment

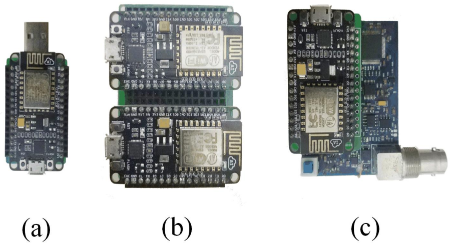
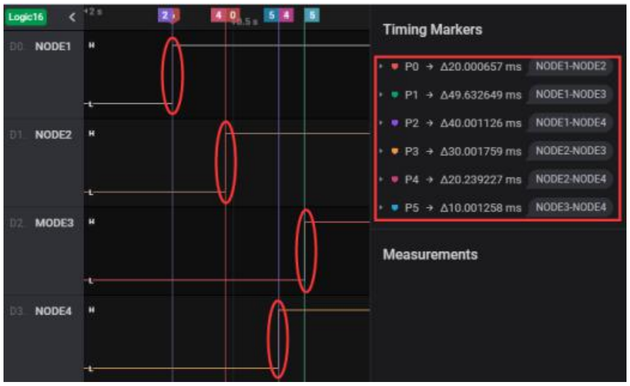

(a) (b) (c)

FIGURE 10. (a)Sink node; (b) Router node; (c)Acquisition node

The experiment environment is shown in Fig. 9. The drive-train diagnostics simulator (DDS) simulates the actual working environment. The gear boxes are used to transfer power and generate vibrations. The motor is used to adjust the gear rotating speed to change the vibration frequency. The vibration sensor collects simple machine vibration data. The acquisition node is the control core, which is responsible for converting and storing data, among other uses. The sink node and assistant, as shown in Fig. 10(a), also have two functions. For the whole system, they are in charge of the network management. For the synchronization trigger, they are used for timing. The routers, as Fig. 10(b) shows, have two basic functions, i.e., transmitting data and establishing the main and adjoint networks. The acquisition node, as shown in Fig. 10(c), offers a timing core. In addition, vibration data are shown on the monitor software.

To compare the accuracy of the path perception synchronization trigger, a set of random tests are conducted without the synchronization method and using the path perception synchronization method. After the formal acquisition command is sent to each node by the terminal assistant, the time sequence of the assigned pins on the ADC of each acquisition node by a logic analyzer is captured. The error is analyzed under these two setups.

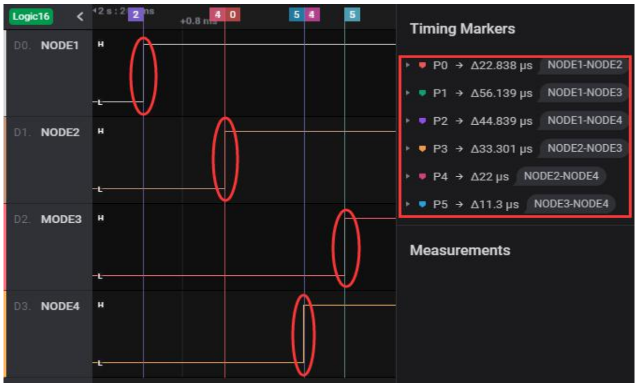


Timing Markers

P0 → Δ20.000657 ms NODE1-NODE2
P1 → Δ49.632649 ms NODE1-NODE3
P2 → Δ40.001126 ms NODE1-NODE4
P3 → Δ30.001759 ms NODE2-NODE3
P4 → Δ20.239227 ms NODE2-NODE4
P5 → Δ10.001258 ms NODE3-NODE4

Measurements

FIGURE 11. Trigger error without synchronization method



Timing Markers

P0 → Δ22.838 μs NODE1-NODE2
P1 → Δ56.139 μs NODE1-NODE3
P2 → Δ44.839 μs NODE1-NODE4
P3 → Δ33.301 μs NODE2-NODE3
P4 → Δ22 μs NODE2-NODE4
P5 → Δ11.3 μs NODE3-NODE4

Measurements

FIGURE 12. Trigger error with synchronization method

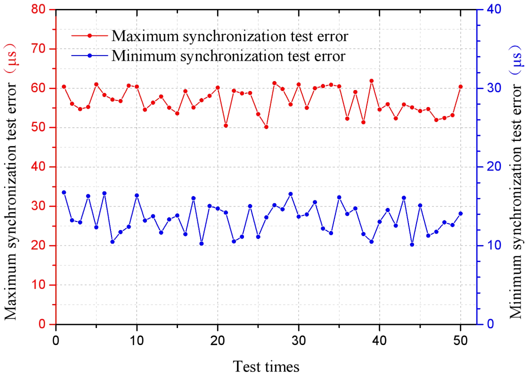


FIGURE 13. Multiple nodes synchronization error test statistics

1) ACCURACY COMPARISON OF EXPERIMENTAL RESULTS

Fig. 11 shows the acquisition error without the synchronization method. The maximum time error of the command trigger among the four acquisition nodes is approximately 49.632649 ms for nodes 1 and 3, whereas the minimum error is 10.001258 ms for nodes 3 and 4. Fig. 12 shows the synchronization error with the path-perception synchronization trigger acquisition method. The maximum time error is approximately 56.139 μs for nodes 1 and 3, whereas the minimum error is 11.3000 μs for nodes 3 and 4. Table 4 shows the compartment of the synchronization error between nodes.

To avoid the contingency caused by the random test, the measurement is repeated 50 times. The maximum and minimum errors are recorded as shown in Fig. 13. The results indicate that the maximum synchronization error is between 50 and 60 μs, and the minimum is between 10 and 16 μs, which meets the requirements of a high-precision synchronization trigger for mechanical vibrations.

2) MODAL TEST AND COMPARISON OF SYNCHRONIZATION TRIGGER

To verify the effectiveness of the path-perception synchronization trigger method, the method is compared with the RBS synchronization algorithm. Owing to the limited number of nodes, the test points were divided into four groups and three test points in each group. During the test, the three accelerometers are spread at different connections with the acquisition node, and one node is responsible for collecting the excitation hammer signals. The excitation point is located at the lower-right corner of the steel frame, and the excitation positions of the four groups are the same. To reduce the experimental error, each group of test points needs to conduct multiple test experiments. Fig. 14 shows a schematic diagram of the steel frame structure, the length, width, and thickness of which are 480, 330, and 45 mm, respectively.

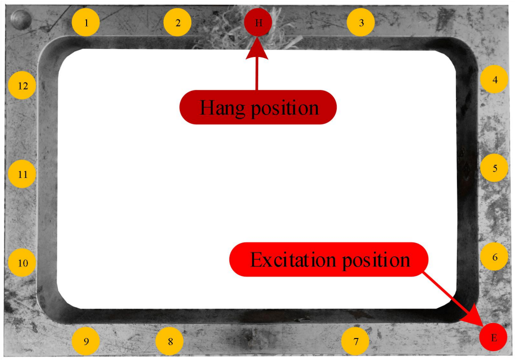
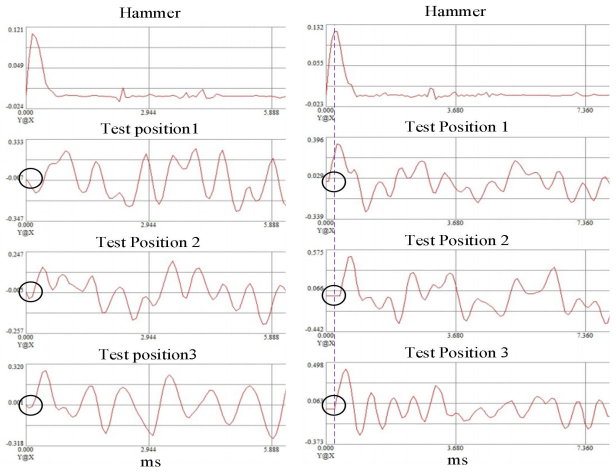


FIGURE 14. Modal test steel frame and test point



(a) (b)

(a)Path precipitation synchronization (b)RBS synchronization

FIGURE 15. Excitation test signal

Fig. 15 shows the enlarged waveform diagram of the modal synchronization test. It is obvious from Fig. 15(a) that each test point triggers the acquisition almost synchronously. It can be seen from the circled part in Fig. 15(b) that the test points 1, 2, and 3 are obviously out of synchronization.

V. CONCLUSIONS

To balance the requirements of a network establishment and synchronization trigger, a new dual-mode multiplexing node design is proposed. Based on this mode, to solve the problems in which the mechanical vibration signal acquired by the wireless sensor relies on beacon synchronization, clock synchronization, and communication between nodes, a path perception time compensation synchronization trigger method based on a clustered network is proposed. Experiment results show that the maximum synchronization error of the multiple-hop synchronization accuracy under the path perception synchronization trigger method is between 50 and 60 μs, and the minimum synchronization error is between 10 and 16 μs. Compared with the non-synchronized method, the accuracy is increased by more than 800 times. It is improved by approximately 6 to 10 times compared to the beacon synchronization and clock synchronization at 300 μs [14, 15]. Finally, the accuracy of the minimum synchronization error is increased by approximately 6 times compared with RBS [18] and TPSN [20]. This indicates that the path perception time synchronization trigger method has an excellent practical effect.

VI. REFERENCES

[1] Y. Guo, N. Wang, Z.-Y. Xu *et al.*, "The internet of things-based decision support system for information processing in intelligent manufacturing using data mining technology," *Mechanical Systems and Signal Processing,* vol. 142, pp. 106630, 2020/08/01/, 2020.

[2] F. Tao, Y. Zuo, L. D. Xu *et al.*, "IoT-Based Intelligent Perception and Access of Manufacturing Resource Toward Cloud Manufacturing," *IEEE Transactions on Industrial Informatics,* vol. 10, no. 2, pp. 1547-1557, 2014.

[3] C. Qian, Y. F. Zhang, C. Jiang *et al.*, "A real-time data-driven collaborative mechanism in fixed-position assembly systems for smart manufacturing," *Robotics And Computer-Integrated Manufacturing,* vol. 61, Feb, 2020.

[4] B. Cheng, S. Zhao, S. Wang *et al.*, "Lightweight Mashup Middleware for Coal Mine Safety Monitoring and Control Automation," *IEEE Transactions on Automation Science and Engineering,* vol. 14, no. 2, pp. 1245-1255, 2017.

[5] C. Du, S. Gao, N. Jia *et al.*, "A High-Accuracy Least-Time-Domain Mixture Features Machine-Fault Diagnosis Based on Wireless Sensor Network," *IEEE Systems Journal,* vol. 14, no. 3, pp. 4101-4109, 2020.

[6] S. A. Putra, B. R. Trilaksono, M. Riyansyah *et al.*, "Intelligent Sensing in Multiagent-Based Wireless Sensor Network for Bridge Condition Monitoring System," *IEEE Internet of Things Journal,* vol. 6, no. 3, pp. 5397-5410, 2019.

[7] Y. Peng, W. Qiao, L. Qu *et al.*, "Sensor Fault Detection and Isolation for a Wireless Sensor Network-Based Remote Wind Turbine Condition Monitoring System," *IEEE Transactions on Industry Applications,* vol. 54, no. 2, pp. 1072-1079, 2018.

[8] M. C. Pan, W. C. Chu, and D.-D. Le, "Adaptive angular-velocity Vold–Kalman filter order tracking – Theoretical basis, numerical implementation and parameter investigation," *Mechanical Systems and Signal Processing,* vol. 81, pp. 148-161, 2016/12/15/, 2016.

[9] Y. Fu, K. Mechitov, T. Hoang *et al.*, "Efficient and high-precision time synchronization for wireless monitoring of civil infrastructure subjected to sudden events," *Structural Control & Health Monitoring*, 2020.

[10] B. Bengherbia, M. Ould Zmirli, A. Toubal *et al.*, "FPGA-based wireless sensor nodes for vibration monitoring system and fault diagnosis," *Measurement,* vol. 101, pp. 81-92, 2017/04/01/, 2017.

[11] Q. Huang, B. Tang, and L. Deng, "Development of high synchronous acquisition accuracy wireless sensor network for machine vibration monitoring," *Measurement,* vol. 66, pp. 35-44, 2015/04/01/, 2015.

[12] X. Xiao, B. Tang, and L. Deng, "High accuracy synchronous acquisition algorithm of multi-hop sensor networks for machine vibration monitoring," *Measurement,* vol. 102, pp. 10-19, 2017/05/01/, 2017.

[13]J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in Proceedings of the 5th symposium on Operating systems design and implementation (Copyright restrictions prevent ACM from being able to make the PDFs for this conference available for downloading), Boston, Massachusetts, 2002, pp. 147–163.

[14] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timing-sync Protocol for Sensor Networks," *SenSys'03: Proceedings of the First International Conference on Embedded Networked Sensor Systems*, 09/16, 2004.

[15] M. Maróti, B. Kusy, G. Simon *et al.*, *The Flooding Time Synchronization Protocol*, 2004.

[16] S. Raza, A. A. Quintana, A. Agnihotri *et al.*, "Dual-Mode Time-Slotted (DMTS) MAC Protocol for Industrial Control Applications." pp. 1-6.

[17]S. Kim, S. Pakzad, D. Culler *et al.*, *Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks*, 2007.

[18] Z. Wang, J. Cao, Z. Cai *et al.*, "Periodicity and finite-time periodic synchronization of discontinuous complex-valued neural networks," *Neural Networks,* vol. 119, pp. 249-260, 2019/11/01/, 2019.

[19] Y. Lv, "Research on address shortage in a tree based ZigBee network." pp. 70-74.

[20]M. Zielinski, F. Mieyeville, D. Navarro *et al.*, *Design of a low power wireless sensor network node for distributed active vibration control system*, 2014.

Hao Fu was born in Chongqing, China, in 1994. He received B.E. degree in 2017 and the M.A. degree in mechanical engineering from the College of Mechanical Engineering, Chongqing University, Chongqing, China, in 2020.

His research interests include wireless sensor networks for mechanical equipment monitoring and edge computing for micro wireless sensor network nodes.

Baoping Tang received the M.Sc. and Ph.D. degrees in mechanical engineering, in 1996 and 2003, respectively, both from the College of Mechanical Engineering, Chongqing University, Chongqing, China.

He is currently a Professor and Ph.D. Supervisor with the College of Mechanical Engineering, Chongqing University. He has authored or co-authored more than 150 papers during his research career. His research interests include wireless sensor networks, mechanical and electrical equipment security services and life prediction, and measurement technology and instruments.

Dr. Tang was the recipient of the Second Prize of the National Scientific and Technological Progress of China in 2004 and the Second Prize of the National Invention of China in 2015.

Yi Huang was born in Chongqing, China,in 1994. He received the B.E. degree in mechanical engineering from the College of Mechanical Engineering, Chongqing University, Chongqing, China, in 2017. He is currently working toward the Ph.D. degree in mechanical engineering under the supervision of Prof. Baoping Tang at the State Key Laboratory of Mechanical Transmission, Chongqing University.

His research interests include wireless sensor networks for mechanical equipment monitoring and energy-harvesting powered wireless sensing nodes.

Lei Deng received her M.Sc. degree in 2001 and Ph.D. degree in 2010 both from Chongqing University. Now she is an associate professor in Chongqing University. Her main research interests include logistics and supply chain management, wirel