
Shiu Kumar1*, Yeonwoo Lee2, and Seong Ro Lee1

1 Department of Information Electronics Engineering, Mokpo National University
2 Dept. of Information and Communication Engineering, Mokpo National University
shiu748@gmail.com, ylee@mokpo.ac.kr, srlee@mokpo.ac.kr

Abstract. A significant challenge when implementing time synchronization protocols is minimizing timestamp uncertainties. The major problem of time synchronization is not only that this packet delay exists, but also being able to predict the time spent on each can be difficult. This paper aims to reduce these uncertainties by estimating them in the transmitter and receiver thus greatly increasing the performance of the time synchronization technique.

Keywords: Time synchronization, packet delay, timestamp

1 Introduction

Wireless sensor networks have drawn much attention from the academia to the industry in recent years [1] due to changes in electronics and communication technologies, and are becoming an increasingly popular research topic. Low cost, multi-functionality, small size and mobility are the advantageous characteristics of WSN that make it suitable for applications such as health monitoring [2], agriculture [3], surveillance [4], etc.

Synchronization is typically based on some sort of message exchange among sensor nodes. However, these message exchanges that take place during synchronization are subject to numerous delays. Such latency depends on the amount of time taken from the moment when the sender timer is read to the instant when the receiver timer is updated. The total latency in terms of end-to-end single-hop transmission includes the following terms: send time, access time, propagation time and receive time.

The RBS protocol is based on the receiver-receiver synchronization method that synchronizes a cluster of wireless sensors within transmission range of the reference sensor node [5]. TPSN [6] is a timing synchronization algorithm for sensor networks based on the sender-receiver synchronization comprising of two phases: level discovery phase and synchronization phase. To mention, a few other time synchronization methods reported are Pairwise Broadcast Synchronization, Pairwise Lightweight Protocol, and Flooding Time Synchronization Protocol.
2 Packet Delay Estimation (PDE)

The proposed PDE method is based on the sender-receiver synchronization that employs a single radio message, and provides multi-hop synchronization. The wireless sensor nodes used in this research comprise of the single chip 2.4GHz Nordic NRF24L01+ radio module in conjunction with the 16MHz Arduino Mega 2560 micro-controller.

Carefully analyzing and compensating for non-deterministic delays known as the packet delay in the radio message delivery will result in higher synchronization accuracy. The timing sequence diagram for the NRF24L01+ radio given in Fig. 1 was thoroughly studied and followed in order to form equations that can be used to estimate the packet delay components.

\[ \text{PD}_T = \frac{1000}{62.5} \left( T_{UL} + T_{CE} + T_{stdby2a} + T_{OA} + T_{IRQ} + T_t \right) \]

where \( \text{PD}_T \) refers to the transmitter packet delay in number of clock cycles, \( T_{CE} = 21.12 \mu s \) is the time for which the PTX CE pin is low and \( T_t \) refers to the software processing time of the transmitter. From the results, the equation to estimate the value of \( T_{UL} \) (\( \mu s \)) was obtained as given by (2) where \( PL \) is the payload size.

\[ T_{UL} = 2.06(PL) + 20.48 \]

The value of \( T_{stdby2a} \) is given to be 130 \( \mu s \) while \( T_{IRQ} \) for air data rate of 1 Mbps is given as 8.2 \( \mu s \) [7]. From the results, (3) was obtained which is used to estimate the value of \( T_{OA} \) in \( \mu s \).

\[ T_{OA} = 4(PL) + 32.65 \]
The proposed system computes (1) after which the time stamping is done. After the time stamping, the transmitter packet delay is added to the timestamp TCNT1 element given by (4).

\[ TCNT1 = TCNT1 + PD_T \]  

(4)

The packet delay components that need to be considered for the receiver are the propagation time, the time taken to validate the received message and the receive time. The time difference between the PTX IRQ and the PRX IRQ denoted as \( T_{IRQ} \) gives the propagation time together with the time taken to validate the received message. Time stamping on the receiver side is done as soon as the received message is validated and available in the PRX FIFO, which is when the PRX IRQ interrupt is generated.

This interrupt pin was used to generate an interrupt on the MCU to read the time the message is received. However, time \( T_{INT} \) is taken for the MCU interrupt to be generated from the time the PRX IRQ pin interrupt is generated. After the time stamping, the MCU reads the message from the PRX FIFO. Once the payload data containing the transmitter time stamp \( T_{tt} \) is read, another time stamp is taken to obtain the time taken to receive the message. The difference between these two PRX time stamps is the receive time, \( T_{recv} \). Hence the total receiver packet delay in number of clock cycles is given by (5), where \( T_r \) is the software processing time of the receiver.

\[ PD_R = \frac{1000}{62.5} (T_{IRQ} + T_{INT} + T_{recv} + T_r) \]  

(5)

Finally, the receiver time is synchronized to this new value of \( T_{tt} \), which accounts for both the estimated transmitter and receiver packet delays.

### 3 Results and Discussion

The PDE method was implemented on wireless sensor nodes comprising of the NRF24L01+ radio module together with Arduino Mega 2560 as its MCU. The PDE algorithm was implemented on two nodes acting as the sender and receiver. Algorithms were written to output a high at one of the MCU outputs at every timer1 overflow interrupt, which is after every 4ms. Low level program, bitSet and bitClear, was used to output high and low at the output pins. This was done for both the sender and receiver in order to measure the synchronization accuracy.

The average synchronization per hop accuracy of the proposed PDE method is 2.20 µs. The best case was the two nodes getting perfectly synchronized and the worst case synchronization accuracy was 4.64 µs. The results are summarized and compared with other methods in Table 1. During the testing, there were interferences from Wi-Fi and other radio modules operating in the laboratory. The synchronization accuracies are also subject to the clock drift as clock skew compensation has not been implemented during testing.
Table 1. Comparison of PDE with most representative clock synchronization protocols in WSNs

<table>
<thead>
<tr>
<th></th>
<th>PDE</th>
<th>TPSN</th>
<th>RBS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average error (in µs)</td>
<td>2.20</td>
<td>16.9</td>
<td>29.1</td>
</tr>
<tr>
<td>Worst case error (in µs)</td>
<td>4.64</td>
<td>44.0</td>
<td>93.0</td>
</tr>
<tr>
<td>Best case error (in µs)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

3 Conclusion

This paper proposed a novel clock synchronization scheme (PDE) for WSNs by minimizing the packet delay uncertainties and attained an average precision of 2.20 µs in the single hop scenario with worst case error of 4.64 µs were shown by providing the experimental results. The proposed scheme only requires one-way timing message exchange between a pair of sender and receiver nodes to achieve network wide synchronization, which significantly reduces the overall energy consumption and communication overhead for achieving global synchronization compared to other synchronization methods.

Acknowledgments. The work was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2011-0029321) and Priority Research Centers program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2009-0093828).

References