A Strategy for Checking Timing Constraint Violation in Wireless Sensor Networks

Jaecheol Kim, Juil Kim, Woojin Lee, and Kiwon Chong
Department of Computing
Soongsil University
Sangdo-5Dong, Dongjak-Gu, Seoul
KOREA

Abstract: - This paper proposes a strategy for checking timing constraint violation in wireless sensor networks. We present the process for checking timing constraint violation and the techniques for calculation of clock drift, collection of sending and receiving time, and calculation of data transfer time. Through the techniques of this paper, the user can know about data transfer time on each node and total data transfer time from a sensor node to the server. Then, the user can confirm whether the expected data transfer time is satisfied by analyzing the information. If the expected data transfer time is not satisfied in the sensor network, the user can find out where the delay has occurred. Real-time data processing is a key factor in sensor networks, so the sensing data should be transferred to the server in right time in order to perform the correct action at proper time. The proposed strategy will help the user to maintain correct real-time processing in the sensor network because the user can debug timing constraint violation.

Key-Words: - debugging, timing constraint, violation checking, sensor networks

1 Introduction

According to the advancement in wireless communications and electronics, low-cost, low-power, multifunctional sensor nodes have been developed. These tiny sensor nodes are composed of sensing, data processing, and communicating components, so they can make sensor network [1]. A sensor network [2, 3] is a wireless network which is composed of a large number of lightweight, low-powered sensor nodes. The sensor networks can be used for various application areas such as health, military, home, and environment monitoring.

Real-time data processing is a key factor in sensor networks. To perform the correct action at proper time, the sensing data should be transferred to the server in time. Accordingly, it is very important to confirm whether the nodes in a sensor network transmit the sensing data to the server in right time.

In this paper, we propose a strategy for checking timing constraint violation in wireless sensor networks. Section 2 describes the process for checking timing constraint violation. We propose the techniques for performing each step in the proposed process in section 3. We discuss the techniques for calculation of clock drift, collection of sending and receiving time, and calculation of data transfer time. We conclude, and present future works in section 4.

2 The Process for Checking Timing Constraint Violation

Figure 1 presents the overall process for checking timing constraint violation in the sensor network.

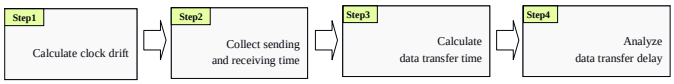

Fig.1. The process for checking timing constraint violation

In step 1, clock drift of each node in the sensor network is calculated. Each node uses its own timer, so clock drift between nodes can be occurred. To calculate correct data transfer time, it is necessary to know clock drift of each node. The clock drift of each node is calculated on the basis of server clock. In step 2, sending and receiving time of data packets are collected. To transfer a data packet from a sensor node to the server, several nodes should send and receive the packet because there are a lot of nodes in the sensor network. It is necessary to collect sending and receiving time of a data packet on each node in order

to calculate data transfer time. In step 3, data transfer time is calculated using sending and receiving time of a data packet on each node. In step 4, data transfer delay is analyzed. The measured data transfer time is compared with the expected data transfer time. The user can confirm whether the expected data transfer time is satisfied. If the expected data transfer time is not satisfied in the sensor network, the user can find out where the delay has occurred.

3 The Method to Check Timing Constraint Violation

In this section, the method for checking timing constraint violation is described. We propose the techniques for performing each step in the process presented in section 2.

3.1 Calculation of Clock Drift

In this paper, clock drift of each node is calculated on the basis of server clock using existing time synchronization techniques [4, 5]. The clock drift is basically calculated using sender-receiver synchronization [6] technique.

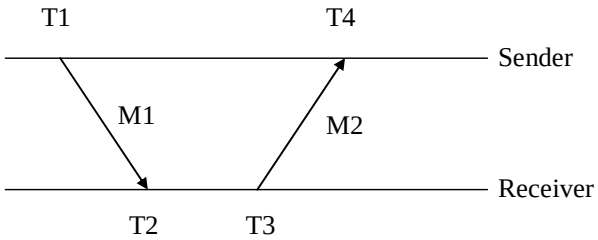

Fig.2. Message exchange between two nodes

Figure 2 presents the message exchange between sender and receiver. T1 is message sending time of sender, and T4 is message receiving time of sender. T1 and T4 is measured using sender's timer. M1 is a message sent by sender in order to get clock drift between sender and receiver. M2 is an acknowledgement message sent by receiver. The clock drift between sender and receiver is calculated through formula (1).

$$D = \{(T2-T1) - (T4-T3)\} / 2 \quad \text{(D: clock drift)} \quad (1)$$

In the proposed method, there are some assumptions to calculate clock drift of sensor nodes. The assumptions are the following.

- ✓ Each sensor node has its own clock.
- ✓ There are cluster heads in the sensor network.
- ✓ There is a central server connected to the sink node.
- ✓ The standard time is local time of the sink node.

Figure 3 presents the method to get clock drift of each node in this paper. Calculation of clock drift is composed of two phases. First, clock drift between the sink node and the cluster header node is calculated. The sink node directly communicates with the server, so clock of sink node is considered as base clock. Secondly, clock drift between the cluster header node and the sensor node is calculated. Through the two phases, clock drift between the sink node and the other node is calculated.

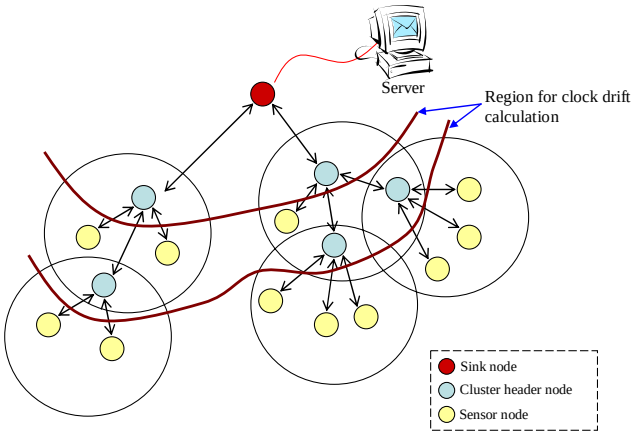

Fig.3. The method to get clock drift

3.2 Collection of Sending & Receiving Time

To transfer a data packet from a sensor node to the server, several nodes should send and receive the packet because there are a lot of nodes in the sensor network. Transfer time from a sensor node to the server is needed for debugging timing constraint violation. Accordingly, it is necessary to collect sending and receiving times of a data packet on each node in order to calculate total transfer time of a data packet from a sensor node to the server.

Figure 4 shows the structure of data packet which is sent to router nodes from sensor nodes. *Node ID* is identification number of a sensor node. Each sensor node can be identified using *Node ID*. *Packet Type* is the type of data packet transmitted by the sensor node. *Data Length* is the number of sensing units which are equipped in the sensor node. A sensor node can equip several sensor units in order to sense various kinds of

data such as gas, light, temperature, and humidity. *Data Length* is necessary to extract sensing data from the packet because data packet is variable according to the number of sensing units. *Sensor Type* is the type of a sensing unit. Several *Sensor Type* fields are included in the packet if several sensing units are equipped in the node. *ADCH* and *ADCL* is the value of sensing data. *Sensor Type*, *ADCH*, and *ADCL* fields are added in the packet according to the number of sensing units. *Send Time* means packet sending time of the node.

Node ID	Packet Type	Data Length	Sensor Type	ADC High	ADC Low	…	Send Time	Padding

- Node ID : Identification number of sensor node (1 byte)
- Packet Type : Type of packet to transmit (1 byte)
- Data Length : Number of sensing unit (1 byte)
- Sensor Type : Type of sensing unit (1 byte)
- ADC High : High byte of ADC value (4 byte)
- ADC Low : Low byte of ADC value (4 byte)
- … : Variable according to the number of sensing unit (variable)
- Send Time : Message sending time (6 byte)

Fig.4. The structure of data packet to send from a sensor node to a sink node or a router node

Figure 5 shows the structure of data packet which is sent to another node from a router node. *Node ID* is identification number of a router node. *Packet Type* is the type of data packet transmitted by the router node. *Dest. Node ID* is identification number of destination node. *Dest. Node ID* can be another router node ID or sink node ID. *Hop Count* is the number of hops. Whenever sensing data is transmitted to a router node, the number of hops is increased by one. *Router Node ID* is identification number of previous router node. *Router Node ID* is same value as *Node ID* if *Hop Count* is 1. *Router Node ID* field is added in the packet whenever data packet is transmitted to a router node. *Send Time* means packet sending time of the node. *Recv Time* means packet receiving time of the node. *Original Data* is data packet which is received from a sensor node.

Node ID	Packet Type	Dest. Node ID	Hop Count	Router Node ID	…	Send Time	Recv Time	Org Data

- Node ID : Identification number of router node (1byte)
- Packet Type : Type of data packet to transmit (1 byte)
- Dest. Node ID : Identification number of destination node (1 byte)
- Hop Count : Number of hop (1 byte)
- Router Node ID : Identification number of previous router node (1 byte)
- … : Variable according to the hop count (variable)
- Send Time : Message sending time (6 byte)
- Recv Time : Message receiving time (6 byte)
- Original Data : Received data (variable)

Fig.5. The structure of data packet to send from a router node to a sink node or a router node

Data sending and receiving times are collected using the two types of packet presented in figure 4 and 5. The sending and receiving times are stored in the database of the server. In the server, data transfer time is calculated using the sending and receiving time.

3.3 Calculation of Data Transfer Time

To calculate data transfer time, the data packet received from each node should be analyzed. In the server, information for calculating data transfer time is extracted from the data packet, and stored in the database. Followings are the information which is stored in the database for calculating data transfer time.

- **Node ID**: Node ID is identification number of each node.
- **Clock drift**: Clock drift is the difference between the server clock and the node clock on the basis of the server clock.
- **Sensing data**: Sensing data is data value which is sensed by a sensor node. Nodes send and receive a large number of data in the sensor network, so transfer time for each data should be calculated.
- **Sending time**: Sending time is packet sending time of the node. Sending time is measured using the timer of the node.
- **Receiving time**: Receiving time is packet receiving time of the node. In the sensor node, receiving time is not checked because sensor node only sends sensing data to another node. Receiving time is measured using the timer of the node.
- **Receiving order**: Receiving order is the data receiving order of the node. Total data transfer time of the sensing data is calculated by summing data transfer time on each node according to the receiving order.

Figure 6 presents the data transfer time between two nodes.

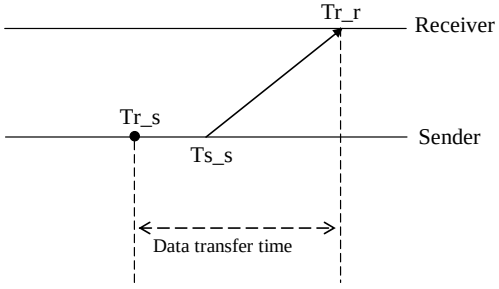

Fig.6. Data transfer time

In the figure, Tr means data receiving time, and Ts means data sending time. In the sensor node, Tr means data sensing time. Tr and Ts are measured using the timer of each node. Accordingly, clock drift is used for calculating data transfer time. The data transfer time between two nodes is calculated using formula (2).

Clock drift of Sender: Ds
Clock drift of Receiver: Dr
Data receiving time of Sender: Tr_s
Data sending time of Sender: Ts_s
Data receiving time of Receiver: Tr_r
Data transfer time: T

$$T = (Ts\_s - Tr\_s) + \{(Tr\_r - Dr) - (Ts\_s - Ds)\} \quad (2)$$

Total data transfer time from a sensor node to the server is calculated by summing the data transfer time calculated using formula (2). (Ts_s-Tr_s) is zero if the Sender is a sensor node. Total data transfer time from a sensor node to the nth node can be calculated using formula (3).

Data receiving time: Tr
Data sending time: Ts
Clock drift: D

$$T_{tot} = \sum [(Ts_{(k-1)} - Tr_{(k-1)}) + \{(Tr_{(k)} - D_{(k)}) - (Ts_{(k-1)} - D_{(k-1)})\}] \quad (2 \leq k \leq N) \quad (3)$$

3.4 Analysis of Data Transfer Delay

The measured data transfer time should be compared with the expected data transfer time. The user can confirm whether the expected data transfer time is satisfied. If the expected data transfer time is not satisfied in the sensor network, the user can find out where the delay has occurred. The user can know following information through analysis of data packet.

- **Hop count**: Hop count means the number of router nodes which join in transfer of a sensing data to the server.
- **Data transfer time on each node**: Data transfer time on each node is calculated using formula (2).
- **Data processing time on each node**: Data processing time on each node is calculated using (Ts_s-Tr_s) of formula (2). It means the difference between data receiving time and data sending time.
- **Total transfer time**: Total transfer time is calculated using formula (3). It means the data transfer time of a sensing data from a sensor node to the server.

The user can find out where the data transfer is delayed through the above information if the expected data transfer time is not satisfied.

4 Conclusion and Future Work

Real-time data processing is a key factor in sensor networks. To perform the correct action at proper time, the sensing data should be transferred to the server in time. Accordingly, it is very important to confirm whether the nodes in a sensor network transmit the sensing data to the server in right time.

In this paper, we have proposed a strategy for checking timing constraint violation in wireless sensor networks. We have presented the process for checking timing constraint violation. The process is composed of four steps. In step 1, clock drift of each node in the sensor network is calculated. In step 2, sending and receiving time of data packets are collected. In step 3, data transfer time is calculated using sending and receiving time of a data packet on each node. In step 4, data transfer delay is analyzed. We have also proposed the method for debugging timing constraint violation. We have presented the techniques for calculation of clock drift, collection of sending and receiving time, and calculation of data transfer time.

Through the techniques of this paper, the user can know about data transfer time on each node and total data transfer time from a sensor node to the server. Then, the user can confirm whether the expected data transfer time is satisfied by analyzing the information. If the expected data transfer time is not satisfied in the sensor network, the user can find out where the delay has occurred.

In the future, we will implement the proposed techniques in order to practically use the techniques. Furthermore, we will develop a debugger for debugging timing constraint violation in wireless sensor networks using the techniques of this paper.

References:

[1] I.F.Akyildiz, W. Su et al., "A Survey on Sensor Networks," *IEEE Communications Magazine*, August 2002.

[2] Dong-Hyun Chae, Kyu Ho Han, Kyung Soo Lim, and Sun Shin An, "Trend and Technology of Sensor Network," *Korea Information Science Society Communications*, VOL.22, NO.12, pp.5-12, 2004.

[3] Shigeru Fukunaga et al., "Development of Ubiquitous Sensor Network," *Oki Technical Review October 2004/Issue 200*, Vol.71 No.4, pp. 24-29, 2004.

[4] Ganeriwal, S., Kumar, R., and Srivastava, M. B., "Timing-Sync Protocol for Sensor Networks," *The First ACM Conference on Embedded Networked Sensor System (SenSys)*, 2003.

[5] Jin Hong No and Young Sik Hong, "Clock Synchronization in Wireless Embedded Applications," *Journal of Korea Information Science Society: Information Networking*, VOL.32, NO.6, pp.668-675, 2005.

[6] D. L. Mills, "Internet Time Synchronization: The Network Time Protocol," *IEEE TRANSACTIONS ON COMMUNICATIONS*, VOL.39, NO.10, pp.1482-1493, 1991.