IEICE TRANS. INF. & SYST., VOL.E95–D, NO.1 JANUARY 2012

LETTER *Special Section on Trust, Security and Privacy in Computing and Communication Systems*

Delay Attack-Resilient Clock Synchronization for Wireless Sensor Networks*

Eui-Jik KIM[†], ***Member***, **Jeongsik IN**[††], **Sungkwan YOUM**[††], ***Nonmembers***, ***and*** **Chul-Hee KANG**[†a)], ***Member***

SUMMARY This paper presents the design and performance evaluation of a delay attack-resilient clock synchronization scheme (abbreviated to DARCS) for wireless sensor networks. In order to provide both synchronization accuracy and robustness, we propose a novel three-way handshake-based protocol, which completely excludes non-deterministic factors such as random backoff durations and unexpected hardware interrupts in a software manner and, in this way, the node can accurately estimate the relative clock offset and the end-to-end delay between a pair of nodes. Consequently, DARCS makes it possible to correct time synchronization errors as well as to detect delay attacks precisely. The simulation results show that DARCS achieves a higher synchronization accuracy and is more resilient to delay attacks than the other popular time synchronization schemes.

key words: *time synchronization, security, delay attack, wireless sensor networks*

1. Introduction

Time synchronization is crucial for many operations and applications of wireless sensor networks (WSNs), such as time slot allocation in TDMA, duty cycle scheduling in CSMA, environmental monitoring, object tracking, and so on [1]. The related works can be classified into two main approaches: receiver-receiver synchronization (RRS) and sender-receiver synchronization (SRS). RBS [2] and FTSP [3] belong to the former type, and TPSN [4], LTS [5], and TS/MS [6] to the latter. The RRS approach uses a "third party" for synchronization, where the nodes in the network synchronize their clocks with the time of arrival by receiving the reference beacon broadcasted from this third party node. In the SRS approach, a pair of nodes exchange the timing message in a two-way handshake pattern when they want to synchronize with each other. Noh et al. [12] proposed a third synchronization approach, called pairwise broadcast synchronization (PBS). PBS is a hybrid version of the above two approaches, in which a pair of super nodes exchange timing messages in an SRS manner, and the nearby neighbors overhear them and estimate the time offset and skew. Its extensions [13], [14] are included in this approach.

Manuscript received March 22, 2011.
Manuscript revised June 24, 2011.

[†]The authors are with the Dept. of Electrical Engineering, Korea University, Seoul, Korea.
[††]The authors are with Samsung Electronics Co., Ltd., Suwon, Korea.

*This work was supported by the IT R&D program of MKE/KEIT [KI001855, Wireless Local Area Communication Systems on Tera Hertz Band]

a) E-mail: chkang@korea.ac.kr (Corresponding author)

DOI: 10.1587/transinf.E95.D.188

These existing schemes commonly have the following limitations when it comes to meeting the requirements of general WSN applications. First of all, because of the uncertainty of the packet delay, due to such factors as the Tx-FIFO buffer loading time, random backoff, and unexpected hardware interrupts, which act as non-negligible sources of error in the calculation of the clock offset, the desired level of time accuracy cannot be achieved. Even if the buffer loading time is assumed to be deterministic, the random backoff that is performed after time stamping the packet at the MAC is highly variable and, thus, would completely overshadow other delays in practice. Moreover, a hardware interrupt occurring before the transmission of the packet would severely degrade the performance of the time synchronization procedure. In order to minimize the repercussions resulting from these non-deterministic factors, FTSP [3] and TPSN [4] realize MAC-layer time stamping in their hardware, i.e. the Mica Atmel (AVR) MCU platform [7], where the node can write its clock time on the timing packet from the PHY to the MAC layer when it is about to be transmitted from the radio module. However, this MAC-layer time stamping based time synchronization is both hardware-dependent and MAC/PHY-layer dependent. Thus, it is not applicable to WSNs with other hardware platforms and cannot be a general solution for WSNs.

Second, none of the existing schemes was designed with security in mind. Thus, they cannot be applied as is to applications in hostile environments (such as a military battlefield or security monitoring area), where security is critical. In [8], Song et al. identified the four possible attacks that may occur in a WSN environment, i.e., the masquerade attack, replay attack, message manipulation attack, and delay attack. The first three attacks can be addressed by cryptographic techniques. However, to prevent a delay attack, specific countermeasures are needed. Ganeriwal et al. [9] proposed a secure pairwise synchronization protocol, where the node can detect pulse-delay attacks by comparing the calculated end-to-end delay with the maximal estimated delay. However, as mentioned above, since the calculated delay includes uncertainties resulting from such factors as random backoffs and hardware interrupts, the solution provided by these techniques suffers from detection errors caused by these uncertainties.

In this paper, we propose a delay attack-resilient clock synchronization scheme (abbreviated to DARCS) which is a software-based solution and is applicable to general WSN nodes regardless of the kind of hardware mote platform. In

DARCS, the node captures the outgoing or incoming time of the timing packets and then provides its counterpart with this information instead of using MAC-layer time stamping. In this way, DARCS can thoroughly exclude the uncertainties caused by random backoff and hardware interrupts, resulting in high precision performance. The nodes in the network can obtain a precisely calculated relative offset and end-to-end delay, by means of which they can effectively detect delay attacks from malicious nodes.

In what follows, we describe the design and performance of the proposed scheme in detail.

2. Delay Attack-Resilient Clock Synchronization

2.1 Calculation of Clock Offset and End-to-End Delay

In DARCS, the node captures the outgoing or incoming time of the timing packet and then sends a packet including this information to its counterpart. For this purpose, the end of start of frame delimiter (SFD) of the packet can be used as the reference point in time. When the node sends or receives a packet, it first senses the preamble followed by the SFD. At the end of the SFD, the timer of the node captures the local clock time and remembers it. Note that, contrastingly, in the existing schemes, the node writes its transmission time on a packet before loading it into the TxFIFO, provided that MAC-layer time stamping is not supported in a hardware manner. Consequently, DARCS can thoroughly exclude timing errors caused by random backoff durations and unexpected interrupts. Due to this salient feature, DARCS uses a three-way handshake protocol. We focus on achieving instantaneous synchronization between the local clocks of the two nodes. Therefore, the goal of the proposed scheme is to estimate the relative offset precisely, so that the local clocks can be corrected accordingly. We suppose that drift or skew errors due to the variation of the oscillator frequency can be corrected by running our scheme periodically.

Figure 1 schematically illustrates the synchronization operation of DARCS in detail. We define the two nodes, namely the reference node and the target node, and the latter synchronizes its local clock with that of the former. When the synchronization procedure starts to run, the reference node first sends a *Sync1* message (empty packet) to the target node after loading it into TxFIFO and waiting for a random backoff duration. At that time, it remembers its outgoing time, $T_{Ref}(s1)$. The target node receives *Sync1* from the reference node writes the incoming time, $T_{Targ}(r1)$, on the *Sync2* message and then responds with it. Similarly, it remembers its outgoing time, $T_{Targ}(s2)$. The reference node receives this *Sync2* message and remembers the incoming time, $T_{Ref}(r2)$, and calculates the end-to-end delay for the forward direction, $FD = T_{Ref}(s1) - T_{Targ}(r1) = \delta + D_{Forward}$, where δ is the relative offset between the nodes and $D_{Forward}$ is the end-to-end delay of the message for the forward direction. Then, the reference node writes the received $T_{Ref}(r2)$ and the calculated FD on the *Sync3* message and sends it to the target node. Now, the target node can calculate the

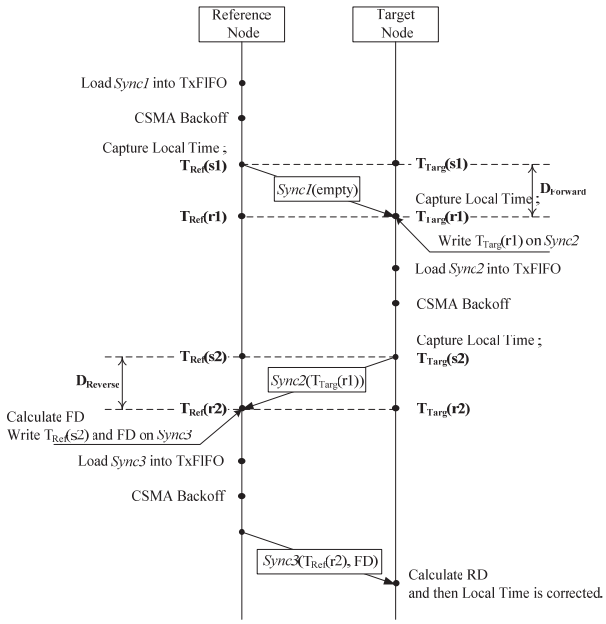

Fig. 1 The operation of time synchronization.

end-to-end delay for the reverse direction, $RD = T_{Targ}(s2) - T_{Ref}(r2) = -\delta + D_{Reverse}$. Note that the values of $D_{Forward}$ and $D_{Reverse}$ do not contain any unknown delay factors, such as random backoffs and unexpected interrupts, at all. Thus, it is reasonable to suppose that $D_{Forward}$ and $D_{Reverse}$ are the same, and δ and D ($= D_{Forward} = D_{Reverse}$) can be calculated as follows.

$$\delta = \frac{(FD - RD)}{2}, \quad D = \frac{(FD + RD)}{2} \tag{1}$$

2.2 Delay Attack Detection

To detect a delay attack on the timing messages, DARCS compares the end-to-end delay, D (calculated from Eq. (1)), and the expected delay, D_{Exp}. Note that the calculation of D is an auxiliary benefit of the operation of DARCS in the security context. We make use of this result in the detection of delay attacks from malicious nodes, even though it is not used in the procedure of time synchronization. On hardware platforms that do not support MAC-layer time stamping, the end-to-end delay calculation of [9] necessarily involves nondeterministic factors, such as random backoff durations and interrupt delays. Thus, it is difficult to detect delay attacks correctly.

The expected delay D_{Exp} is defined as the maximal one-hop propagation delay. In this paper, this value is assumed to be known in advance. It is estimated from the RSSI value when exchanging a packet between a pair of nodes or obtained by measurements performed on real hardware motes. Consequently, after completing the three-way handshaking procedure, the target node calculates the end-to-end delay, D, first, and then determines whether the current transaction is a delay attack or not by comparing it with D_{Exp}. If D is greater than D_{Exp}, the node makes the decision that the

current transaction is a delay attack. By applying this security feature, we complete our time synchronization solution, which reconciles the two objectives of robustness and time accuracy.

3. Performance Evaluation

We evaluate the performance of the proposed scheme through experimental simulations using the MATLAB simulator, and compare it with TPSN [4] and TS/MS [6]. TPSN is the first SRS-based protocol. Apart from TS/MS, most SRS-based protocols commonly use a similar two-way handshake procedure to that of TPSN, in which the relative offset and end-to-end delay are estimated based on exchanged timestamps. Meanwhile, TS/MS uses a different method of single-hop synchronization, in which the timestamps are used to establish "bounds" on the relative drift and offset. For convenience of evaluation, in the experiment of TS/MS, the offset is determined as the median value of the established offset bound. The parameters used in the simulation refer to the IEEE 802.15.4 specification [10] and CC2420 datasheet [11]. We assume that the protocols run on a general hardware platform, which does not support MAC-layer time stamping. We further assume that the size of the timing packets is 50 bytes and D_{Exp} is set to 1.76 *msec*, which has a margin of 10% on the normal one-hop propagation delay. The random backoff duration is considered as a non-deterministic factor, which causes uncertainty in the synchronization procedure. For this, we define the metric, the contention rate, in which 0.1 denotes a backoff range of [0, 40] *msec*. In the simulation, an intentional offset value is randomly added at intervals of 1 *msec* and the synchronization procedure is continuously repeated for the purpose of intuitively observing the error-compensation performance. Note that, in a real environment, such an additional offset occurs due to the relative drift of the clock frequency after the initial offset correction and is even smaller than the intentional offset used in the simulation. Thus, it can be inferred that DARCS would exhibit better performance in a real environment.

Figures 2 (a) and (b) show the time differences between the reference node and target node, *viz.*, the synchronization error as a function of time when the contention rate is 0.05 and 0.15, respectively. In these figures, for the periodically added offset errors, the clock of the node using DARCS is quickly corrected to a time difference of 0, while the nodes using TPSN and TS/MS do not compensate their clocks correctly. Since, in DARCS, the nodes exchange their outgoing or incoming times instead of using time stamping, they are hardly affected by non-deterministic factors. A slight increase in the delay is induced when exchanging the timing packet in three-way handshaking. In contrast, TPSN and TS/MS cannot reduce the effect of random backoff in a software manner on a general platform. Thus, as the contention level increases, their performance in terms of improving the synchronization precision is severely degraded. This can also be inferred from the results in Figs. 3 and 4.

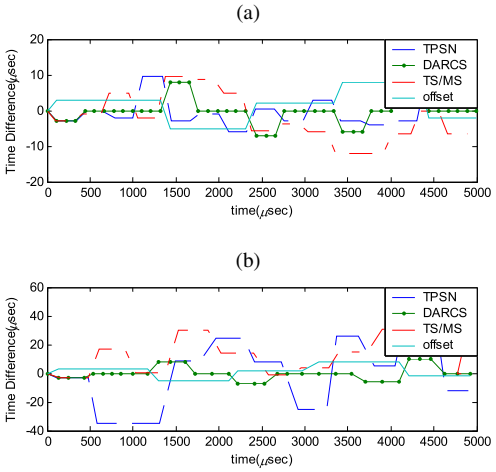

Fig. 2 The time difference as a function of the time.

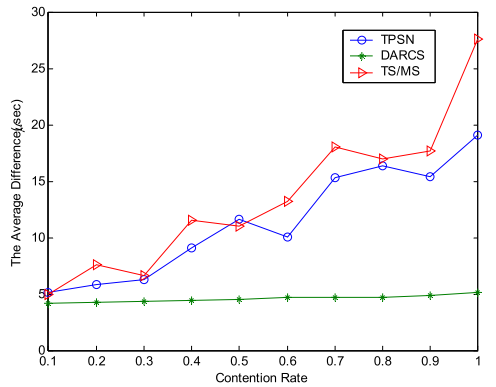

Fig. 3 The average time difference as a function of the contention rate.

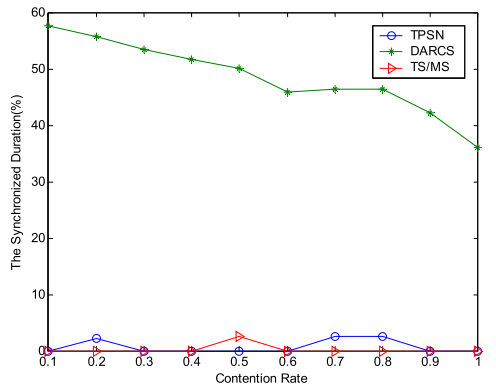

Fig. 4 The synchronized duration as a function of the contention rate.

As discussed above, DARCS provides the precise estimation of the end-to-end delay as well as the relative offset, in a manner which is independent of error-prone factors due to the network conditions, e.g. the contention level and unexpected interrupts. Knowledges of the precise value of the end-to-end delay enables us to detect delay attacks by malicious nodes. Figure 5 shows the fault detection rate for delay attacks, which represents the ratio of misidentified normal transactions to delay attacks. In the figure,

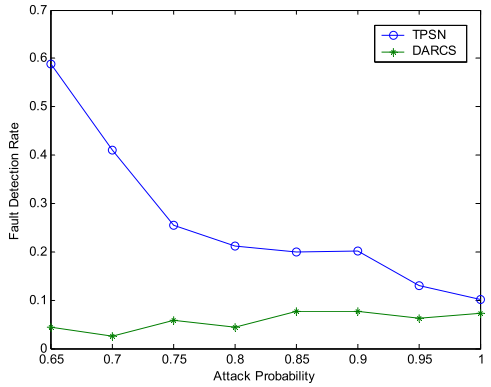

Fig. 5 The fault detection rate as a function of the attack probability.

TS/MS is not considered for the performance comparison, since it does not support the estimation of the end-to-end delay. On the whole, DARCS exhibits a lower fault detection rate than TPSN. If the hardware platform does not support MAC-layer time stamping, the end-to-end delay estimated by TPSN includes error-prone factors, such as interrupt handling and the channel access time, which can have an effect on the decision as to whether a delay attack is in progress. Therefore, a number of normal transactions might be misidentified as an attack. On the other hand, DARCS can precisely estimate the propagation delay by exchanging timestamps at the application-layer in a 3-way handshake manner, which thoroughly excludes such error factors. Note that the problem of unknown delay variance can be addressed by the tuning of the experimental margin associated with the one-hop delay.

4. Conclusion

This paper presents DARCS, which is a delay attack-resilient clock synchronization scheme for wireless sensor networks. To reconcile the two objectives of robustness and accuracy, DARCS involves the following salient features: 1) A three-way handshake-based protocol, which enables the time offset and end-to-end delay to be precisely estimated by exchanging the incoming/outgoing times of a timing message without using hardware-dependent MAC-layer time stamping, and 2) Delay attack detection, which provides the exact level of detection for delay attacks by malicious nodes with a minor fault rate. The performance evaluation shows that the proposed scheme has better accuracy than the existing protocols and effectively defends against delay attacks.

References

[1] I.F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," IEEE Commun. Mag., vol.40, no.8, pp.102–114, 2002.
[2] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," SIGOPS Operating Systems Review, vol.36, issue SI, Dec. 2002.
[3] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The flooding time synchronization protocol," SenSys '04, The 2nd international conference on Embedded networked sensor systems, pp.39–49, Nov. 2004.
[4] S. Ganeriwal, R. Kumar, and M.B. Srivastava, "Timing-sync protocol for sensor networks," SenSys '03, The 1st international conference on Embedded networked sensor systems, pp.138–149, Nov. 2003.
[5] J.V. Greunen and J. Rabaey, "Lightweight time synchronization for sensor networks," WSNA '03, The 2nd ACM international conference on Wireless sensor networks and applications, pp.11–19, Sept. 2003.
[6] M.L. Sichitiu and C. Veerarittiphan, "Simple, accurate time synchronization for wireless sensor networks," WCNC '03, Wireless Communications and Networking, pp.1266–1273, 2003.
[7] Mica2 and Mica2Dot, http://www.xbow.com/Products/Wireless_Sensor_Networks.htm
[8] H. Song, S. Zhu, and G. Cao, "Attack-resilient time synchronization for wireless sensor networks," MASS '05, IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 2005.
[9] S. Ganeriwal, C. Popper, S. Capkun, and M.B. Srivastava, "Secure time synchronization in sensor networks," Transactions on Information and System Security (TISSEC), vol.11, no.4, pp.23–35, July 2008.
[10] IEEE Std 802.15.4 2006, Wireless Medium Access Control and Physical Layer Specifications for Low-Rate Wireless Personal Area Networks, IEEE Computer Society, June 2006.
[11] CC2420 Datasheet, www.ti.com
[12] K.-L. Noh, E. Serpedin, and K. Qaraqe, "A new approach for time synchronization in wireless sensor networks: Pairwise broadcast synchronization," IEEE Trans. Wireless Communications, vol.7, no.9, pp.3318–3322, Sept. 2008.
[13] K.-L. Noh, Y.-C. Wu, K. Qaraqe, and B. Suter, "Extension of pairwise broadcasting clock synchronization for multi-cluster sensor networks," EURASIP J. Advances in Signal Processing, special issue on Distributed Signal Processing Techniques for Wireless Sensor Networks, vol.2008.
[14] K.-Y. Cheng, K.-S. Lui, Y.-C. Wu, and V. Tam, "A distributed multihop time synchronization protocol for wireless sensor networks using pairwise broadcast synchronization," IEEE Trans. Wireless Communications, vol.8, no.4, pp.1764–1772, April 2009.