Node Identification Using Clock Skew

Ibrahim Ethem Bagci, Utz Roedig

School of Computing and Communications, Lancaster University, Lancaster, UK
{i.bagci, u.roedig}@lancaster.ac.uk

Abstract. Clocks on wireless sensor nodes experience a natural drift. This clock skew is unique for each node as it depends on the clocks manufacturing characteristics. Clock skew can be used as unique node identifier which is, among other applications, useful for node authentication. We describe how clock skew of a node's clock can be measured directly on a node by utilising the available high precision radio transceiver clock. We detail an implementation of this proposed local clock skew tracking method for the Zolertia Z1 platform. We determine the required sampling effort to accurately determine clock skew. We also discuss how clock skew measurements can be aligned with existing transceiver operations in order to avoid an increase in energy consumption.

1 Introduction

All clocks on wireless sensor network (WSN) platforms experience a natural drift. This drift is unique to a node as it depends on the clock hardware. For example, the drift of a node's real-time clock is defined by unique properties of the used quartz crystal. For most WSN applications clock drift is a nuisance and mechanisms such as time synchronisation protocols are put into place to combat it. However, in this paper we take advantage of a node's unique clock drift pattern and use it to uniquely identify nodes.

Besides other security requirements it must be possible to authenticate nodes and sensor data provided by them. For example, a sink node must be able to verify that data is provided by genuine nodes and not by an adversary. Classical cryptographic methods can be used to implement authentication. In this case shared keys are used to identify nodes. As keys may become compromised (i.e. someone obtaines a copy) which would allow an adversary to impersonate a node, methods have been developed which bind authentication to a node's hardware. For example, a crypto chip such as the Atmel ATSHA204 [1] can be included in a node's design which holds cryptographic material for authentication. In this case an attacker must obtain the crypto chip in order to impersonate a node. However, crypto chips are expensive and require an additional component to be integrated in the node design. Thus, we use an already present hardware characteristic to derive a unique node identification; we use a node's unique clock skew characteristic for identification.

In addition to the outlined security application, clock skew based identification is useful for other tasks. For example, when commisioning nodes unique identifiers

such as node IDs and communication addresses must be determined. Clock skew can also be used in this broader context to generate unique identifiers.
Clock skew has been previously used as means of node identification. For example, Kohno et al. [2] have shown that clock skew is unique and can be used to identify classical PCs in the Internet. Uddin et al. [3] have shown that this method can be used in principal in the context of wireless sensor networks. Existing work determines clock skew by comparing clocks on separate nodes (or nodes and a sink). For this process it is necessary to distribute time stamps over the underlying communication network and a *constant* network delay is required. In a WSN context this is an impractical requirement as duty cycled communication induces large network delay variances.
In our work we move away from this limitation of existing work and we measure clock skew locally on nodes. We discuss how this can be achieved in general, describe an implementation of this method and provide a detailed analysis describing clock sampling requirements and achievable quality. More specifically, the contributions of this paper are:

- *Local Skew Determination:* We describe how the clock skew of a node's crystal-based real-time clock can be measured locally using the high precision clock available on modern transceivers to create unique node identifiers. An implementation of this method for the Zolertia Z1 platform using Contiki is described.
- *Analysis of Sampling Requirements:* The achievable quality of the local clock skew determination is compared with the quality of state of the art distributed methods. The dependency of clock sampling effort and skew calculation quality is analysed in detail. In this context it is also shown how clock sampling can be aligned with general transceiver operation in order to avoid a transceiver duty cycle increase.

The remaining paper is organised as follows. The next Section describes related work. Section 3 defines the term clock skew formally and methods for clock skew calculation and analysis are discussed. Section 4 discusses remote clock skew determination while Section 5 describes local clock skew determination. In Section 6 we discuss findings and describe future directions of research work.

2 Related Work

Clock skew is the deviation of a clock from the true time. Fingerprinting devices using clock skew is carried out by comparing frequencies of two clocks, one of them generally assumed to represent the true time.
Kohno et al [2] has shown that clock skew of devices can be measured to fingerprint devices. It is shown that the clock skew of each device is unique and stays fairly consistent over time.
Zander et al. [4] improved clock skew measurement by applying a technique called synchronized sampling. They demonstrated that synchronization of samples reduces the quantisation error and hence improves skew determination quality.

Jana et al. [5] used clock skew to fingerprint wireless devices. The motivation for their work was the detection of fake wireless access points. Their work demonstrates that emitting timestamped beacons with high frequency allows for precise clock skew calculation. According to their observations 50 to 100 beacons are sufficient to estimate clock skew accurately enough to identify individual nodes.
Arackaparambil et al. [6] demonstrated a clock skew spoofing attack in 802.11 networks by using virtual interfaces. In their work they propose methods to combat clock skew spoofing and propose standardised interfaces which would allow network providers to publish clock skew information.
Huang et al. [7] demonstrated clock skew-based identification in wireless sensor networks in the context of the Flooding Time Synchronization Protocol (FTSP) [8]. FTSP provides coarse estimation of clock skew based on current offset and previous skew (it uses linear regression on the past 8 data points).
Uddin et al. [3] demonstrated that sensor nodes have a unique clock skew and that the clock skew of a node can easily be monitored.
Murdoch et al. [9] split skew into two components, a constant and a variable part. The variable part is affected by temperature changes and this effect was used to reveal node identities in the TOR network by influencing CPU load and hence the temperature of devices leading to measurable clock skew changes.
Our work differs from existing approaches as we measure clock skew locally on nodes. We believe that this is a necessary step towards a practical system as variations in communication delays cannot be avoided in any real-world WSN deployment. Furthermore, we show how clock skew measurements fit with energy efficient operations of sensor nodes and investigate the required sampling effort in detail. Existing work with the exception of Huang et al. [7] calculate clock skew offline after a long sequence of samples are collected using a linear programming approach. Huang et al. [7] calculate clock skew online using a linear regression approach which we adopt in our work.

3 Clock Skew

Clock skew could be determined by analysing the drift of a clock C_m with the help of a stable reference clock C_r. However, in a practical setting a stable reference clock is generally not available and it is only possible to monitor one drifting clock with another drifting clock. Hence, a measured clock skew value for a node reflects drift of the measured clock and the used reference clock. Nevertheless, the determined clock skew value is unique and dependant on the hardware characteristics of the clocks used.

3.1 Definition of Clock Skew

The measured clock C_m runs at frequency f_s and the reference clock C_r runs at frequency f_r. To determine clock skew, timestamps of the measured clock and the reference clock are taken periodically. T_1^m and T_1^r are the first timestamps of both clocks taken at the first sample point, T_i^m and T_i^r are timestamps taken

at the i^{th} sample point. The elapsed time of the measured clock C_m at the i^{th} sample point is $t_i^m = (T_i^m - T_1^m)/f_m$; the elapsed time at the reference clock C_r is $t_i^r = (T_i^r - T_1^r)/f_r$. The *offset* – the difference between measured and reference clock – at the i^{th} sample point is $o_i = t_i^m - t_i^r$. If we sample N pairs of $(t_i^r,\, o_i)$ for $i \in \{1, ..., N\}$ and plot these pairs (the so called offset-set), we obtain an approximately linear graph. It is possible to fit a linear function of the form

$$\delta \cdot t_N^r + \varphi \tag{1}$$

to these obtained measurement points. The slope δ of the fitted linear function is called the *clock skew*.

3.2 Clock Skew Determination

There are a number of methods available to fit a linear function. Depending on the exact method used the definition of *clock skew* is also slightly altered. In the literature mainly two methods are used in the context of clock skew calculation which we detail next and use in the remainder of the paper.

Linear Programming - LP Moon et al. [10] have shown that clock skew can be accurately estimated from a set of samples using a linear programming method. LP finds a line $\delta \cdot t_i^r + \varphi$ that upper bounds the offset-set. The problem constraint of LP is given as

$$\delta \cdot t_i^r + \varphi \geq o_i \tag{2}$$

and the following function is minimized (object function):

$$\frac{1}{N} . \sum_{i=1}^{N} (\delta \cdot t_i^r + \varphi - o_i) \tag{3}$$

LP delivers accurate results but has drawbacks when considering a WSN context. Samples must be collected and stored before the calculation can begin. Furthermore, a relatively complex solver for the LP must be available. If considering to execute the calculation on a resource constrained node storage and calculation requirements may be too excessive.

Linear Regression - LR Maróti et al. [8] estimate clock skew using a form of linear regression in their Flooding Time Synchronisation Protocol (FTSP). The algorithm uses the average elapsed time at the reference clock $\overline{t^r}$ and the average offset $\overline{o}$ up to the i^{th} sample point. Then the skew is estimated by calculating:

$$\delta = (o_i - \overline{o})/(t_i^r - \overline{t^r}) \tag{4}$$

For the FTSP 8 sample points are used to estimate the clock skew. We use this algorithm with a varying number of sample points in order to have control over the achievable quality of the skew estimation. Compared to LP the implementation is much simpler and therefore more suitable for usage on resource constrained nodes.

3.3 Clock Skew Quality

In this work we aim to use clock skew to uniquely identify nodes. Hence it is important that clock skew measurements on two nodes can be attributed to the individual nodes. The collision probability (the likelihood that two nodes are seen as the same even though they are different) should be as small as possible. Clock skew measurement, for example using the previously described LP or LR method, is subject to variation. Thus, as two nodes may have a skew values close to each other, variance of the measured skew may make it hard to clearly attribute measurements to individual nodes.
We use a t-test to compare the means of the measured clock skews of two nodes. The t-test returns a test decision for the null hypothesis that the nodes are the same. The alternative hypothesis is that the two nodes are not the same. The probability value (p-value) returned by the t-test is the probability of wrongly assuming that the two analysed nodes are the same when they are in fact not. Therefore, a small p-value indicates that the means of clock skew measurements of two investigated nodes are unlikely to be the same.
For the experimental evaluations described in later sections we use p-values to describe quality of determined clock skew measurements.

4 Remote Clock Skew Determination

In existing work (for example, [3] and [7]) measured clock C_m and reference clock C_r are located on different nodes in the network. For example, the reference clock is the real-time clock of the sink node and the measured clock is the real-time clock of the node to be identified. This remote clock skew determination is carried out as a node generally provides only one accurate clock in its default configuration which is suitable for skew calculations.

4.1 The Impact of Network Jitter

As shown in Section 3, timestamps T_i^m and T_i^r are the i^{th} sample taken at the same point in time. When using remote clock skew determination T_i^m is taken first and the timestamp is transmitted via the network to the reference node which then takes the corresponding time stamp T_i^r. Thus, T_i^r is taken Δ_i after the sample T_i^s. Δ_i is the network delay associated with transmitting the i^{th} timestamp of the measured clock. The clock skew calculation as presented in Section 3 is still valid if the network delay is constant ($\Delta_i = \Delta \forall i$). Variations in Δ_i (network jitter) reduce the quality with which the clock skew can be derermined. In a practical WSN network jitter is high due to the nature of duty cycled communication. For example, when using a protocol such as ContikiMAC [11] the forwarding delay is dependant on when a sender transmit request occurs relatively to the point in time when a receiver node enters its periodic listen phase. Using standard ContikiMAC configuration settings, forwarding delays vary between $0ms$ and $125ms$ (ContikiMAC uses channel check rate of $8Hz$). This is too high to measure the characteristics of clock skew.

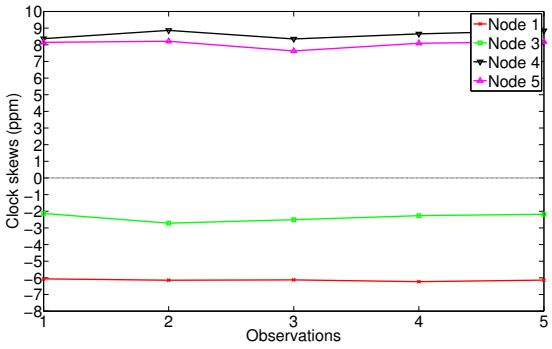


Fig. 1: The measured clock skew using five Zolertia Z1 nodes with remote clock skew determination. 5 observations are carried out using 2500 timestamp samples.

Existing work uses remote clock skew determination in specific setups where nodes are only one hop distance away and no duty cycling MAC protocols or significant network traffic is present. In such a specific case network jitter is low and remote skew determination is possible. However, in any practical setting this method has its limitations.

4.2 Experimental Evaluation

We use five Z1 nodes from Zolertia [12] to carry out a baseline experiment using remote clock skew determination. We use the results obtained in this experiment as comparison for the local clock skew determination method we introduce later. The Z1's external $32,768Hz$ watch crystal is used for clock skew profiling. We label the nodes from Node 1 to 5 and select Node 2 as the sink node. Nodes are

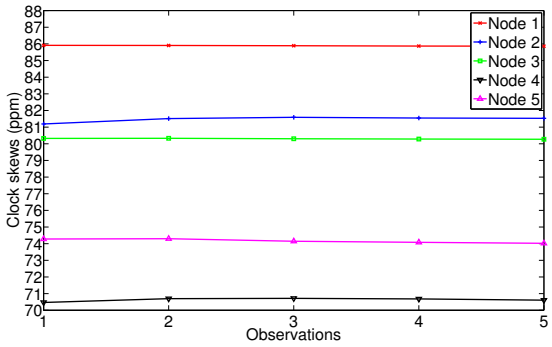


Fig. 2: The measured clock skew using five Zolertia Z1 nodes with local clock skew determination. 5 observations are carried out using 600 timestamp samples.

	Node 1	Node 3	Node 4	Node 5
Node 1	-	$4.25E-06$	$3.19E-08$	$2.36E-08$
Node 3	$4.25E-06$	-	$3.91E-07$	$1.69E-07$
Node 4	$3.19E-08$	$3.91E-07$	-	$3.23E-03$
Node 5	$2.36E-08$	$1.69E-07$	$3.23E-03$	-

Table 1: Obtained p-values describing how clearly nodes can be distinguished from each other node. The smaller the value the more clearly nodes are distinguishable.

Node 1	Node 3	Node 4	Node 5
$1.14E-01$	$9.43E-01$	$4.49E-01$	$3.54E-01$

Table 2: Obtained p-values when comparing a node with itself. Values are larger (2 magnitudes) then the ones shown in Table 1, indicating skews are not distinguisable.

one hop away from a sink node and its external crystal clock is used as reference clock C_r. The nodes run the Contiki operating system and use a NullMAC (no dutycycling MAC is present). No other network traffic than the transmission of timestamps from the 4 measured nodes is present in the network.
Timestamps T_i^m are transmitted every $4s$ from the nodes to the sink. When the sink node receives a timestamp T_i^m it records the corresponding timestamp T_i^r. The total transmitted packet count is 2500 for every node. We repeat this operation 5 times (5 observations) and estimate the clock skew for each node. The clock skew is estimated using the previous described LP method. This calculation is carried out offline after collection of all timestamp samples. The result is shown in Figure 1; clock skew is shown in "part per million" (ppm).
Figure 1 shows that nodes can be clearly identified by measuring clock skew. The clock skew is stable enough over several observations. For some nodes it is easier to distinguish them (for example, Node 1 and Node 4 are clearly separate nodes), others have skew values close together (for example, Node 4 and Node 5). However, even though some nodes are close together measured skew values can be clearly attributed to nodes. How clearly nodes can be identified as separate (the quality of the skew values) can be expressed using the previously outlined t-test. Table 1 shows the results of this analysis. To provide some means of judging p-values we provide Table 2. Here, the means of the first two observations of a node are compared with the mean of the third and forth observation of a node; effectively p-values are generated where a node is compared with itself and values are generated that are not distinguishable. As it can be seen, worst case p-values in Table 1 are two orders of magnitude lower then in Table 2. This indicates that the investigated set of nodes in this experiment is clearly uniquely identifiable via measured clock skews.

5 Local Clock Skew Determination

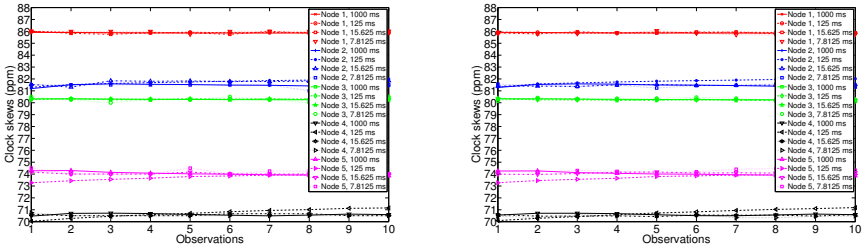
To overcome the previously outlined limitations of remote clock skew determination (i.e. the need of a constant network delay), it would be beneficial to use two local clocks on a node for skew determination.

5.1 Local Clock Sources

Most sensor node platforms provide two clock sources, the crystal-based real-time clock and a processor internal digitally controlled oscillator (DCO). However, these two available local clocks cannot be used for skew determination as the DCO clock has a much lower precision then the real-time clock. Thus, no stable clock skew values can be determined using this setting.
However, most node platforms have a radio transceiver chip which has internally a high precision clock which is necessary for timing of data transmissions. In most cases it is possible to access this resource and use it within the platform for other purposes than transmission and reception of data.
The Zolertia Z1 platform we use for our work provides an $8MHz$ clock within the CC2420 radio transceiver. We use the same approach that Pettinato et al. [13] used to make use of the clock source of the radio. The Clear Channel Assessment (CCA) pin of the radio is alternatively configured to output the internal radio clock signal [14]. We connect the radio CCA pin with the external timer sources pin (TBCLK) of the MSP430 processor. This allows us to use the radio clock as alternative clock source. As this clock is of better quality than the crystal based RT clock it is now possible to use these two clocks to determine locally on the node a stable clock skew value.

5.2 Experimental Evaluation

In this experiment we use the same five Z1 nodes as used for the previously described baseline experiment. The Z1's external $32,768Hz$ watch crystal (the measured clock C_m) is used again for clock skew profiling. The CC2420 radio clock source is used as reference clock C_r. Obviously, all nodes have their individual transceiver clock and, thus, for each node skew profile an individual reference clock is used. Timestamps T_i^m and T_i^r are collected with an interval of $1s$. The results determined using the LP method are shown in Figure 2.
As it can be seen, nodes are clearly identifiable in terms of observed clock skew. The skew values are different to the values obtained remotely as shown in Figure 1. This has to be expected as different reference clocks are used.
The calculated p-values for the five nodes are shown in Table 3. As can be seen here, p-values are several magnitudes lower compared to the remote skew determination. This means that individual nodes can be distinguished much more clearly. It has to be noted that this significant improvement is achieved even though less sample points (600 compared to 2500) are taken and a shorter sample durations (1s compared to 4s) are used.
We conclude that the local clock skew determination is much better then remote clock skew determination as nodes can be more clearly identified. The collision probability (the likelihood that two nodes are seen as the same even though they are different) is greatly reduced. This is an important factor for the design of security mechanisms.



(a) Clock skews using LP.

(b) Clock skews using LR.

Fig. 3: The measured clock skew of five Zolertia Z1 nodes with remote local clock skew determination. 10 independent observations with 600 timestamp samples are used.

5.3 Processing Optimisation

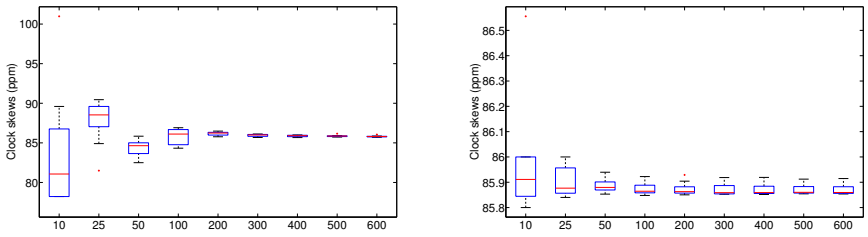
So far we have used the LP method to determine clock skew. This method is useful and provides sufficiently accurate results (as previously shown) for local and remote clock skew determination. However, LP is too complex to execute directly on resource constrained nodes. We therefore use the LR method which is computationally more feasible. LR is expected to deliver results of lesser quality compared to the LP method. It is therefore necessary to analyse if quality of the results is still sufficient to distinguish individual node clock skews.
Figure 3 shows a comparison of local skew determination using the LP and LR skew calculation method (The figure also contains lines showing the effect of reducing the sample period from $1s$ down to $7.8125ms$; we discuss the effect of reduced sample period in the next section). The corresponding p-values are given in Table 3. Interestingly, the LR method fares sightly better in terms of producing clearly distinguishable clock skew values. This is contrary to what we would have expected. Clearly, LR therefore represents a feasible option for clock skew calculation.
We have implemented the LR method of calculation for the Contiki operating system (LP was executed in Matlab after data had been collected). The run-time complexity of the LR algorithm is $O(n)$. For the calculation of skew using 200 sample points an execution time of $2.01ms$ is required which corresponds to an energy consumption of $0.027mJ$.

5.4 Sampling Optimisation

So far we have demonstrated the feasibility of local clock skew determination using relatively long sampling periods of $1s$ and and a relatively large sample size of 600 samples. It is beneficial to reduce as much as possible sample period and sample size. Reducing the sample period and size allows for more energy efficient operations and enables us to obtain skew values faster. When reducing the sample period less time to observe skew is available and resolution of the

	Method	Node 1	Node 2	Node 3	Node 4	Node 5
Node1	LP	-	$1.30E-15$	$1.6E-24$	$8.13E-22$	$6.58E-19$
	LR	-	$1.92E-16$	$6.12E-25$	$1.74E-22$	$3.12E-19$
Node2	LP	$1.30E-15$	-	$1.63E-10$	$8.32E-21$	$1.21E-15$
	LR	$1.92E-16$	-	$1.59E-11$	$8.79E-21$	$1.46E-16$
Node3	LP	$1.36E-24$	$1.63E-10$	-	$4.13E-20$	$8.29E-17$
	LR	$6.12E-25$	$1.59E-11$	-	$7.01E-21$	$3.98E-17$
Node4	LP	$8.13E-22$	$8.32E-21$	$4.13E-20$	-	$2.03E-13$
	LR	$1.74E-22$	$8.79E-21$	$7.01E-21$	-	$4.36E-14$
Node5	LP	$6.58E-19$	$1.21E-15$	$8.29E-17$	$2.03E-13$	-
	LR	$3.12E-19$	$1.46E-16$	$3.98E-17$	$4.36E-14$	-

Table 3: p-values using LP and LR with 10 observations, 600 samples, $1s$ sample period.



(a) Node 1 using a $7.8125ms$ sample period. (b) Node 1 using a $1s$ sample period.

Fig. 4: Node 1 skew for different sample sizes and sample period of $7.8125ms$ and $1s$.

used clocks can be a limiting factor. When reducing the sample size less points are available to reduce variance of the obtained result.

For local clock skew determination as presented in this paper the transceiver clock is required. This clock is only available if the transceiver chip is active. A duty cycled MAC protocol will aim to put the transceiver into an energy efficient sleep state for as long as possible and only wake the transceiver for short durations for transmissions and receptions. If short sample periods are feasible it is possible to align communication and clock sampling and no additional transceiver chip wake times must be introduced. For example, many slotted MAC protocols have transceivers periodically on for durations of $10ms$ to facilitate reception or transmission.

The required sample size should be as small as possible as this would allow us to obtain a skew measurement fast. If we assume that sample periods are aligned with natural transceiver activity we still have to wait until the transceiver was used sufficiently often enough before a skew measurement can be obtained.

We record 600 samples with four different sampling intervals: $1s$, $125ms$, $15.625ms$ and $7.8125ms$. These results are shown for 10 observations in Figure 3, using LP and LR for analysis. Figure 4a and Figure 4b show changes in skew variance of 10 observations when modifying sample size and sample period.

It is clearly visible that skew variations experienced from one measurement to the next are reduced when increasing sample period and/or sample size. The

question is what a feasible combination of sample size and sample period is. The answer will depend on the application situation. However, generally it can be assumed that the shortest feasible sampling period should be used as this helps in aligning sampling with general transceiver activity. Then, the number of samples should be reduced up to a point a sufficient quality (expressed as p-values) of skew calculation is ensured. For example, if we assume that a sample period of $7.8125ms$ is chosen and a sample size of 200 is chosen we obtain p-values as shown in Table 4. These p-values are better than the p-values obtained via remote skew determination. Thus, with these settings local skew detection can replace remote skew detection without a loss in skew quality.

	Node 1	Node 2	Node 3	Node 4	Node 5
Node 1	-	$6.66E-10$	$6.14E-12$	$3.14E-16$	$3.37E-15$
Node 2	$6.66E-10$	-	$1.21E-04$	$3.25E-12$	$2.08E-12$
Node 3	$6.14E-12$	$1.21E-04$	-	$3.79E-12$	$1.92E-11$
Node 4	$3.14E-16$	$3.25E-12$	$3.79E-12$	-	$1.69E-09$
Node 5	$3.37E-15$	$2.08E-12$	$1.92E-11$	$1.69E-09$	-

Table 4: p-values using LR with 10 observations, 200 samples, $7.8125ms$ sample period.

6 Conclusion

A node's unique clock skew can be used for node identification purposes. It is useful to use this approach because node identification is bound to the hardware and no additional components have to be incorporated in the node design. We have demonstrated that clock skew can be determined reliably locally on nodes. Existing methods rely on jitter free network communication which is unachievable in most practical WSN deployments. Thus, the presented work takes an important step towards practical clock skew identification.

We have shown that clock skew of a node's RT clock can be determined locally using the transceiver clock present in most WSN systems. We have shown that locally determined skew values for nodes can be as unique and distinguishable as skew values determined in a distributed fashion. A sample period of $7.8125ms$ and a sample size of 200 is sufficient to determine clock skew locally with the same quality as remotely with a sample period of $4s$ and a sample size of 2500. Also, the possible short sample period of $7.8125ms$ allows us to take clock skew measurements during times the transciever is active for communication. Additional transceiver active periods do not have to be scheduled to achieve local clock skew determination.

Initial experiments have shown that the measured clock skew depends on temperature. A node would need to be profiled in terms of skew over the expected temperature range. Skew values would have to be transmitted together with a temperature reading in order to allow identification in deployments with varying temperature.

References

1. Atmel: Atmel ATSHA204 datasheet (March 2012) http://www.atmel.com/Images/Atmel-8740-CryptoAuth-ATSHA204-Datasheet.pdf.
2. Kohno, T., Broido, A., Claffy, K.: Remote physical device fingerprinting. Dependable and Secure Computing, IEEE Transactions on **2**(2) (2005) 93–108
3. Uddin, M., Castelluccia, C.: Toward clock skew based wireless sensor node services. In: Wireless Internet Conference (WICON), 2010 The 5th Annual ICST. (2010) 1–9
4. Zander, S., Murdoch, S.J.: An improved clock-skew measurement technique for revealing hidden services. In: Proceedings of the 17th conference on Security symposium. SS'08, Berkeley, CA, USA, USENIX Association (2008) 211–225
5. Jana, S., Kasera, S.K.: On fast and accurate detection of unauthorized wireless access points using clock skews. In: Proceedings of the 14th ACM international conference on Mobile computing and networking. MobiCom '08, New York, NY, USA, ACM (2008) 104–115
6. Arackaparambil, C., Bratus, S., Shubina, A., Kotz, D.: On the reliability of wireless fingerprinting using clock skews. In: Proceedings of the third ACM conference on Wireless network security. WiSec '10, New York, NY, USA, ACM (2010) 169–174
7. Huang, D.J., Teng, W.C., Wang, C.Y., Huang, H.Y., Hellerstein, J.: Clock skew based node identification in wireless sensor networks. In: Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE. (2008) 1–5
8. Maróti, M., Kusy, B., Simon, G., Lédeczi, A.: The flooding time synchronization protocol. In: Proceedings of the 2nd international conference on Embedded networked sensor systems. SenSys '04, New York, NY, USA, ACM (2004) 39–49
9. Murdoch, S.J.: Hot or not: revealing hidden services by their clock skew. In: Proceedings of the 13th ACM conference on Computer and communications security. CCS '06, New York, NY, USA, ACM (2006) 27–36
10. Moon, S., Skelly, P., Towsley, D.: Estimation and removal of clock skew from network delay measurements. In: INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. Volume 1. (1999) 227–234 vol.1
11. Dunkels, A.: The ContikiMAC Radio Duty Cycling Protocol. Technical Report T2011:13, Swedish Institute of Computer Science (December 2011)
12. Zolertia: Zolertia Z1 datasheet (March 2010) http://zolertia.com/sites/default/files/Zolertia-Z1-Datasheet.pdf.
13. Pettinato, P., Wirström, N., Eriksson, J., Voigt, T.: Multi-channel two-way time of flight sensor network ranging. In Picco, G., Heinzelman, W., eds.: Wireless Sensor Networks. Volume 7158 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 163–178
14. Texas Instruments: 2.4 GHz IEEE 802.15.4 / ZigBee-Ready RF Transceiver (Rev. C) (March 2013) http://www.ti.com/lit/ds/symlink/cc2420.pdf.