

# Vertical and horizontal synchronization services with outlier detection in underwater acoustic networks

Fei Hu<sup>1</sup>, Yamin Malkawi<sup>1</sup>, Sunil Kumar<sup>2</sup> and Yang Xiao<sup>3\*,†</sup>

<sup>1</sup>*Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY, U.S.A.*

<sup>2</sup>*ECE Department, San Diego State University, San Diego, CA 92182, U.S.A.*

<sup>3</sup>*Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487, U.S.A.*

## Summary

Underwater Acoustic Networks (UANs) have important applications in ocean exploration and lake pollution monitoring. UANs are however different from terrestrial sensor networks due to their highly variable, long propagation delay, and mobility. Clock synchronization is an important protocol to achieve timing-based sensor communications. In this paper, we propose a three dimensional, scalable UAN time synchronization scheme that can achieve both horizontal (i.e., in the same water depth) and vertical (i.e., from bottom up to the surface) clock synchronization to overcome the effects of long acoustic delay. To secure UAN clock synchronization services, we also propose a two-step security UAN synchronization model: (1) correlation test and (2) statistical reputation and trust model. The proposed model can detect outlier timestamp data and identify nodes generating insider attacks. Copyright © 2007 John Wiley & Sons, Ltd.

---

**KEY WORDS:** wireless security; underwater acoustic networks; time synchronization

---

## 1. Introduction

The lakes and rivers impact human life at all levels. Monitoring the water quality is crucial for ensuring safe water supplies. Current, water quality monitoring system mostly uses a small number of water sensors and satellite/cellular communications to collect sensor data remotely. These water sensors are usually expensive, heavy, and consume much energy, due to long-term monitoring and long-distance communications with the onshore stations. Recently the concept of Underwater Acoustic Networks (UANs) [1] has been proposed to implement the next generation water quality monitoring system. Besides chemical material

measurement, the UAN also has important applications in ocean exploration and navy.

UAN typically consists of low-cost water sensors and uses multi-hop wireless transmissions to relay sensed data to a surface station. However, the radio propagation has very limited communication range under water. For example, Mica-2 sensor nodes transmission range has been measured as less than 1 m in fresh water [1]. In fact, even though the long-wavelength Radio Frequency (RF) can penetrate water over longer distance, it requires large transmit power and large antenna. This makes it inappropriate for small, low-power sensor nodes. The underwater acoustic communication provides an important alternative,

\*Correspondence to: Yang Xiao, Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487, U.S.A.

†E-mail: yangxiao@ieee.org

because the acoustic sound travels faster and longer in water than in air [1].

Acoustic UAN systems are significantly different from any ground based sensor network and face some unique challenges. The acoustic signal propagation speed in water is only about 1500 m/s, which is five orders of magnitude lower than radio propagation speed of  $3 \times 10^8$  m/s in the air. As a result, the acoustic signal propagation latency between an underwater transmitter and a receiver that are 2 km apart is comparable to the one between the earth and the moon in radio transmission [2]. This *huge propagation delay* has significant impact on network protocol design. As the huge end-to-end Round Trip Time (RTT) becomes the performance bottleneck, many common network protocols do not work as expected if they are directly ported from radio networks. Moreover, water sensors may fail because of fouling and corrosion, and battery power is limited and usually batteries cannot be easily recharged. Therefore, a delay-tolerant cross-layer communication stack is needed to overcome the effects of long acoustic delays and to minimize energy consumption. Source localization is also necessary to keep track of the exact locations of nodes when contaminants are indicated from the water quality parameters received from the water sensors.

Specifically, *time synchronization* is critical to UAN protocol design. It enables lower duty cycle of the acoustic transducer sending/receiving operations, accurate water sensor localization, and collaborative acoustic signal processing. Note that using lower duty cycles can help in saving energy. More examples of UAN protocols where precise time is needed include, measuring the time-of-flight of the acoustic signals, forming a low-power Medium Access Control (MAC) layer framing schedule [3–5], integrating a time-series of water pH detections into chemical components variance analysis, setting up proper timer expiration when waiting for data fusion from children nodes, ordered logging of chemical data during system debugging, globally coordinating among nodes using consistent clock signals in all the nodes. The time synchronization cannot be achieved by installing Global Positioning System (GPS) systems in UAN nodes due to the following reasons. The high-frequency radio wave used by GPS is quickly absorbed by water, and hence cannot propagate deeply in a UAN platform which uses low transducer power (typically  $< 20$  mW), short-range (50–300 m), and acoustic communications. The UAN time synchronization scheme should therefore use multi-hop, gradual

clock adjustments between sensors at different water depths.

In this paper, we will describe our research results on UAN time synchronization mechanism that is based on a scalable three dimensional (3-D) UAN topology management scheme. We achieve both horizontal (i.e., at the same water depth) and vertical (i.e., from bottom up to the surface) clock synchronization. Our synchronization scheme overcomes the effects of long acoustic delays in UANs. Due to its multi-level clock coordination nature, it can also scale to a large amount of water sensors. Furthermore, due to their importance to public health and ecology analysis, the UANs are potential network attack targets. By attacking the time synchronization services, adversaries can significantly compromise the UAN operations. For instance, (1) the water contaminant source cannot be accurately located; (2) packets will be lost if the sleep-wakeup schedules among neighboring nodes are messed up, and this can further trigger many unnecessary packet retransmissions if the MAC layer acknowledgments are used; and (3) the water quality data cannot be correctly logged into the database due to inaccurate timing information.

UAN time synchronization could possibly face two types of attacks: external and insider attacks. Examples of external attacks include Sybil, Wormhole, Data replay. Typically the adversaries do not have knowledge of the cipher keys or security protocol procedure, which makes the cryptographic prevention schemes effective. Since a water sensor is envisioned to be low-cost, it would not be feasible for manufacturers to make them temper-resistant against an insider attack. An adversary can undetectably take control of the sensor node by physically compromising it. Thus all the keys in the node can be accessed by node capturers, which make all traditional cryptographic schemes (such as one-way hash function) ineffective against an insider attack. Since the traditional cryptographic schemes could overcome typical external attacks, this research will focus on a UAN time synchronization protocol that is able to resist insider attacks through a two-step security model: (1) we first use a correlation-based approach to detect outlier time-stamps, which may indicate a potential insider attack; (2) a long-term-based, statistical trust evaluation procedure is used to identify the real insider attacks.

Related work is summarized as follows. Many time synchronization protocols [6] for ground sensor networks have been proposed recently. However, these schemes do not consider security issues. Only a few recent schemes have considered potential time

synchronization attacks as follows. Some simple attack cases, such as eavesdropping, are summarized in Reference [7]. The symmetric-key based encryptions are suggested in Reference [8] for a multi-hop case. The prevention from delay attacks based on timestamp mapping is discussed in Reference [9]. However, they have not considered many other critical attacks, such as insider attacks, wormhole [10]. Moreover, all of them target general ground sensor networks and ignore many specific characteristics of UANs such as high propagation delays (which can expose to more dangerous attacks such as wormhole), mobile topology due to hidden water currents (which needs efficient re-keying schemes), etc.

This paper is organized as follows. Section 2 presents the assumed UAN network model. Section 3 describes our proposed delay-tolerant UAN vertical and horizontal synchronization mechanism that can achieve satisfactory timestamp accuracy. Section 4 describes our proposed correlation-based security approach to detect outlier timestamp data, and our statistical trust approach details for identifying an insider attacker. Next, detailed experiment analysis is provided in Section 5 to validate the efficiency of our proposed attack-resilient UAN synchronization scheme. Finally, Section 6 concludes this paper.

## 2. UAN Network Model

Our research is based on the following UAN network model [1] that fits the design goal of next generation underwater monitoring systems, shown in Figure 1. Assume that a large number of inexpensive, disposable water sensors are deployed to cover a large area of the lake/ocean. Metal pieces (staying in the bottom of the lake) and thin ropes can be used to tie up sensor nodes in order to distribute them in different desired

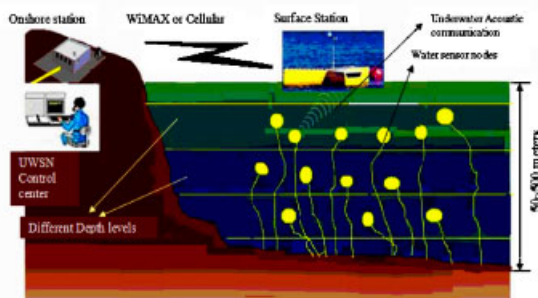


Fig. 1. Targeted UAN system architecture model.

depth levels. The purpose of deploying sensors at different water depths is to better observe water quality parameters. Additionally such a deployment strategy also helps build a tree-like multi-hop routing topology so that the sensor data can be sent to the Surface Station (SS), which floats on the water surface to receive all sensed data. The SS can use long distance radio communication systems, such as Wi-Max or Cellular Networks, to send the data to an onshore station where the monitoring center can perform further data analysis. Note that the acoustic delay is not only large ( $\sim 300$  ms in worst case [1]) but also highly variable due to the unpredictable water currents.

## 3. Vertical and Horizontal Synchronizations

The time synchronization between neighboring nodes at different depths is crucial; otherwise, the clock deviation will accumulate in vertical direction when the synchronization signals propagate from the SS to the underwater nodes. Moreover, data fusion typically occurs between vertical nodes.

Although the effect of clock skew is tiny during radio-based synchronization, it causes inaccuracies which increase with the message propagation latency. In UAN, this error can be significant due to the 50–300 ms of acoustic propagation delay even at moderate distances ( $\sim 250$  m) [2]. We enhance the simple one-hop synchronization scheme discussed in Reference [2] through close integration of the time synchronization procedure with the 3-D, tree-like UAN topology discovery scheme discussed in Subsection 3.2, where SS is the tree root.

### 3.1. Vertical Synchronization

To overcome the clock skew, our scheme consists of periodic Gradual Depth Timing (GDT) phase every  $T_1$  seconds (typically set up to 30 s in a lake with maximum 500 m) and Level 1 (i.e., between the SS and first depth nodes) Skew Compensation (LSC) phase every  $T_2$  seconds which is integer times of  $T_1$ . GDT thus operates more frequently than LSC. Because GDT utilizes tree root as starting point when adjusting the whole UAN timing, it is important to correct Level 1 clock skew/offset first. To save the energy of the SS, the LSC phase should not occur

frequently. The two phases use the same channel for common sensed data transmissions. Typically they use short-range acoustic signal with around 100–200 m of communication distance, at data rate of 20–100 kb/s and the acoustic transducer power < 20 mW [1].

### 3.1.1. LSC phase

Its purpose is to reduce the effects of long acoustic delay on clock skew in Depth 1 nodes. It utilizes the accurate local clock in the SS (since it is not inside water, it may use GPS system and radio communication to periodically obtain accurate universal clock) to adjust the timing in level 1 nodes. The LSC phase requires the SS to broadcast a beacon message to level 1 nodes for  $M$  rounds. Many rounds of timing data can be used for linear regression so as to correct the clock skew; we set up  $M$  as 30 in experiments. The time gap between two broadcasting rounds is  $T_3$ , and we use 100 ms so that its value has the same order as the acoustic delay. The beacon message from the tree root in round  $i$  has the timestamp  $T_{\text{ROOT}}^i$  at the MAC level. Note that lots of non-determination in timing message exchange can be removed by placing message timestamps in the MAC layer instead of in application layer. Suppose that a UAN node  $R$  receives the beacon message at absolute time  $T_{\text{ROOT}}^i + \Delta_{\text{ROOT} \rightarrow R}$ , where  $\Delta_{\text{ROOT} \rightarrow R}$  is the propagation delay between the SS and the node  $R$ . In practice, its value can be approximated by the average one-depth propagation delay and can be easily obtained from empirical experimental data. We model node  $R$ 's uncorrected clock as  $F_R^{\text{no}}(t)$  and its

clock drift of  $R$ 's local clock with respect to the root's reference clock through the linear regression over the following data points with  $X$ -axis and  $Y$ -axis values as:

$$\begin{cases} X\text{-axis} = T_{\text{ROOT}}^i - F_R^{\text{no}}(T_{\text{ROOT}}^i + \Delta_{\text{ROOT} \rightarrow R}) \\ Y\text{-axis} = F_R^{\text{no}}(T_{\text{ROOT}}^i + \Delta_{\text{ROOT} \rightarrow R}) \end{cases} \quad (i = 1, 2, \dots, M) \quad (2)$$

Linear regression attempts to explain a relationship with a straight line fit to  $X$ -axis and  $Y$ -axis values. We use the linear regression to obtain the skew correcting conversion of the local time at node  $R$ . Through the same way all nodes could be skew synchronized with its skew corrected local time, which is represented as  $\tilde{F}_R^{\text{yes}}(t)$ . The following operation will use  $\tilde{F}_R^{\text{yes}}(t)$  to obtain the final skew/offset corrected time  $F_R^{\text{yes}}(t)$ : Based on the classical two-way synchronization exchange [12], after the node  $R$  estimates skew using linear regression, it uses a short-range ( $\sim 150$  m) acoustic signal to send a Sync Request message with skew corrected timestamp  $\tilde{F}_R^{\text{yes}}(T_1^R)$  to the root. Note that this is different from Reference [12] since they just used  $T_1^R$ . The root records its local version of the receiving time  $T_2^{\text{root}}$ , and then returns the value  $T_2^{\text{root}}$  to node  $R$  in a Reply message with timestamp  $T_3^{\text{root}}$ . Suppose that node  $R$  receives the Sync Reply message at

$$T_4^R = \tilde{F}_R^{\text{yes}}(T_3^{\text{root}} + \Delta_{\text{ROOT} \rightarrow R}) \quad (3)$$

We can then calculate the Correction\_factor in Equation (1) as:

$$\text{Correction\_factor} = \frac{\{[T_2^{\text{root}} - \tilde{F}_R^{\text{yes}}(T_1^R)] - [\tilde{F}_R^{\text{yes}}(T_3^{\text{root}} + \Delta_{\text{ROOT} \rightarrow R}) - T_3^{\text{root}}]\}}{2} \quad (4)$$

corrected time as  $F_R^{\text{yes}}(t)$ . Their values can be determined as follows:  $F_R^{\text{no}}(t) = \text{Skew} \bullet t + \text{Offset}$  and

$$F_R^{\text{yes}}(t) = F_R^{\text{no}}(t) + \text{Correction\_factor} \quad (1)$$

where skew/offset is the difference between the time/frequency reported by a clock and the true time/frequency, and frequency is the rate at which the clock progresses.

Then node  $R$  could assign a local time to the absolute time as  $F_R^{\text{no}}(T_{\text{ROOT}}^i + \Delta_{\text{ROOT} \rightarrow R})$ , which has clock error due to the clock skew and offset in addition to the large acoustic propagation delay. Based on the skew theory [11], we could model the

Therefore, when the exchange is complete, the node is able to factor out the skew error due to long acoustic delay and obtain the exact offset.

### 3.1.2. GDT phase

Its purpose is to periodically adjust timing between nodes in neighboring lake depths toward deep lake nodes. During each round, starting from the SS, each parent node exchanges timestamp messages with their children nodes. Each timing message has a unique transaction ID for identifying the GDT rounds. Our GDT depth-by-depth adjustment procedure is based on the improvement of the time diffusion

protocol [13]. Specifically, we have replaced its cluster head selection with a light-weight child tree aggregation algorithm to eliminate the redundant timing adjustment.

### 3.2. Horizontal Synchronization

Practical UAN has a 3-D architecture, and some nodes at Depth  $i + 1$ , shown in Figure 2, may not be able to find parent nodes at Depth  $i$  after the topology self-organization phase, especially when the nodes at Depth  $i$  are not dense. Here, we call those nodes without parent nodes as ‘orphans’, shown as the non-solid nodes in Figure 2. In fact, from time to time, nodes may fail to work due to water corrosion, fish biting, or power exhaustion. After a certain percentage of nodes fail, there will be orphan nodes that cannot obtain timing information from any parent nodes through the above discussed vertical synchronization scheme in Subsection 3.1. Those orphan nodes can only utilize the non-orphan nodes at the same depth to obtain current timing information. We call this procedure as horizontal synchronization. Then horizontal synchronization problem is defined as how we establish a scalable, low-overhead horizontal synchronization mechanism to make the orphan nodes achieve accurate time synchronization with the non-orphan nodes. We call the water sensors at the same depth as ‘brothers’. Our algorithm relies on a distributed 3-D topology discovery scheme. An orphan sensor could use the discovered multi-hop path in the same depth to communicate with other brothers until finally reaching a non-orphan sensor that has at least a parent sensor. After the horizontal connectivity map is established, starting from any non-orphan sensor, the time synchronization between non-orphan nodes and orphan nodes will be conducted periodically with the same period as GDT phase based on the hop-to-hop clock adjustment scheme suggested in the Time Diffusion scheme [13].

Here, we describe our Horizontal Topology Discovery (HTD) procedure that could establish a

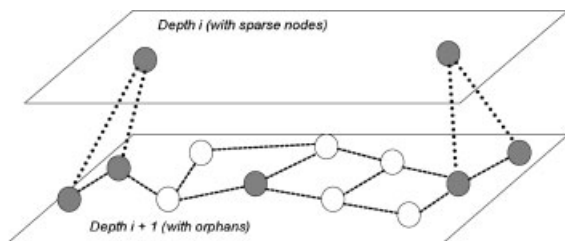


Fig. 2. Orphan nodes (the hollow circles).

shortest multi-hop acoustic path between an orphan sensor and a nearby non-orphan sensor. During HTD an orphan needs to choose a brother node from its neighbors that can provide the minimum number of hops to the SS. This is achieved by exchanging short control packets. Those packets have at least the following two fields in their headers: (1) the energy levels resident in the sensor; and (2) the depth level of each sensor. Each sensor has to maintain two lists: (1) the parent set and (2) the brother set. Obviously an orphan’s parent list is empty. If an orphan is already connected to a neighbor whose parent list not empty, the orphan has to switch to that neighbor’s parent. The two lists are continuously updated as soon as the control packets are received. They become invalid after a preset expiration time. Figure 3 shows the list update procedure.

An orphan node determines a non-orphan sensor with the minimum distance as follows: the orphan first broadcasts an Orphan Help (OH) packet through AODV routing protocol [14]. The OH packet contains the source address, the destination address and the time-to-live (TTL) fields. Any non-orphan that hears this packet replies with Orphan Help Reply (OHR) packet. The first arrived OHR packet defines the brother node to be selected that has the minimum hops. The transmission of the OH/OHR packets is flooded based on AODV scheme. The orphan node stops flooding the OH packets if its parents list is not empty, which means that it has found a parent. The algorithm is illustrated in Figure 4.

An orphan node maintains only a single path to a non-orphan sensor even when multiple paths could be established. To achieve minimum network connectivity among all sensors at the same depth, we have designed a dominant-based topology discovery mechanism to achieve the shortest-distance orphan-to-non-orphan communication [15]. Each non-orphan node serves as a dominant, which is the center of a cluster with orphan nodes as its cluster members. After receiving the horizontal synchronization request from a dominant, a sensor will trigger a hop-to-hop timing propagation process among clusters based on the Time Diffusion scheme [13]. The major horizontal synchronization procedure is explained below.

#### 3.2.1. Dominating-set-based horizontal routing

A subset of the vertices of a graph is called a dominating set if every vertex not in the subset is adjacent to at least one vertex in the subset [15]. The Marking process [16] has been verified to be able to

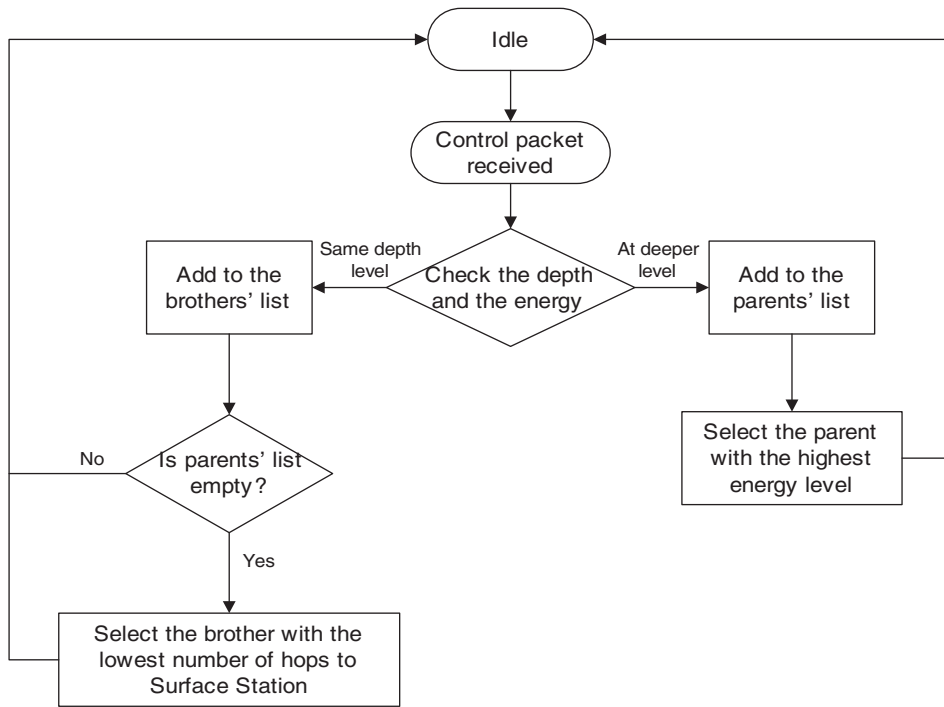


Fig. 3. Parent/bother list update procedure.

efficiently form a dominating set. Assuming that the orphan nodes play the roles of the gateway nodes that are defined in Reference [16], the horizontal orphan-to-non-orphan communication could utilize the dominating-set-based routing protocol that usually consists of three steps [16]: (i) if the source is not a gateway host, it forwards the packets to a source

gateway, which is one of the adjacent gateway hosts; (ii) this source gateway acts as a new source to route the packets in the induced graph generated from the connected dominating set; (iii) eventually, the packets reach a destination gateway, which is either the destination host itself or a gateway that connects the destination host. In the later case, the destination

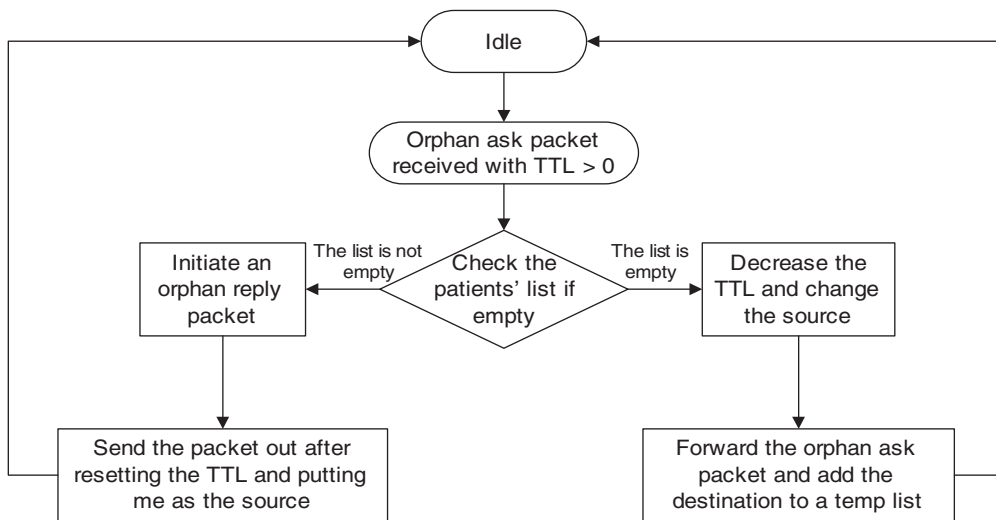


Fig. 4. Searching a closest non-orphan sensor.

gateway forwards the packets directly to the destination host.

### 3.2.2. Horizontal clock synchronization

We can then integrate the above dominating set forming algorithm with the Time Diffusion scheme [13] to achieve Horizontal Synchronization (HS). Its basic procedure is as follows:

The HS algorithm runs in multiple cycles, with cycle duration of  $\tau$ . In each cycle, a non-orphan sensor node initiates a diffusion of timing messages to orphan nodes. This procedure is called Diffused Timing Propagation (DTP). During DTP we need to eliminate outlier nodes, whose clock variance is above some threshold based on a specific type of variance calculation, termed the Allen variance [17]. This variance is determined by exchanging messages and calculating deviations between pairs of adjacent nodes using a Peer Evaluation Procedure (PEP) [13]. The timing information messages are diffused (in hop-to-hop fashion) when they are issued from a non-orphan sensor. Each of such messages contains the following three fields in its header: dominant node's local ID, the number of hops that the timing information message is to be diffused, and the diffused time from the non-orphan node. The orphan sensors at Hop 1 respond to the timing information messages with ACK messages. Afterwards, the round-trip time  $\Delta$  between the non-orphan and orphan nodes is calculated by:  $\Delta(i) = T_1 - T_0$ , where  $T_1$  is the arrival time of the ACK message and  $T_0$  is the broadcast time of the timing information message in cycle  $i$ . Since each orphan node may receive multiple ACK messages, the average of the round trip delays is calculated and used to estimate the one-way delay between the orphan sensor and the neighboring sensors. As a result, the diffused time  $T_{m,i}$  from the orphan node can be calculated as:  $T_{m,i} = T_{M,i} + \frac{\Delta}{2} + \delta$ , where  $\Delta/2$  is the estimated one-way delay, and  $\delta$  is the amount of time that the neighboring sensors wait before adjusting their local clocks. Furthermore, the standard deviation  $\alpha$  of the round trip delays can be obtained and used to estimate the quality of the diffused time  $T_{m,i}$ . Note that a large value of  $\alpha$  indicates that the diffused time may have a larger error. Hence, the standard deviation is accumulated at every hop starting from the orphan node. This accumulated deviation value  $\beta_{m,k}$  can be calculated as  $\beta_{m,k} = \beta_{m,k-1} + \alpha$ , where  $k$  is the distance from the non-orphan sensor in terms of the number of hops.

In summary, in horizontal direction (i.e., the sensors belonging to the same water depth), an orphan sensor at cycle  $i$  could periodically adjust its local clock based on the diffused time  $T_{m,i}$  from a non-orphan sensor, with the consideration of the deviation  $\beta_{m,k}$ . Eventually, the algorithm will converge to a stable status, and any orphan sensor could synchronize with its closest non-orphan sensor.

## 4. Correlation-based Outlier Detection

As mentioned before, many of the 'external' attacks (such as traffic insertion) could be prevented by appropriate cryptographic techniques. However, for 'insider' attacks, cryptography is inadequate to distinguish between inserted data, failed, un-calibrated and real data. A feasible approach is to analyze the timestamp data itself since the ultimate goal of adversaries is to make the timestamp data abnormal. We thus have proposed a window-based outlier detection scheme to detect potential insider attacks based on the following fact (see Figure 5): the distances (and thus the acoustic propagation delays) between any two neighboring acoustic sensors fit a certain probability distribution [18]. Thus there exist strong correlations between the sending and receiving time stamps (their gap reflects the propagation delay). In practice, the statistical distribution of acoustic propagation delays between two neighboring nodes can be obtained beforehand through training experiments. Contaminated timestamp values can be deduced by the correlation-based sliding-window mechanism (see Definition 3), which uses dynamic threshold to detect outlier timestamp values.

**Definition 1. Timestamp Correlation Test:** Suppose that the timestamp information is exchanged between two neighboring water sensors. One of them sends out packets with sequences of sending timestamps  $T_S$ , and the other sensor receives packets at its local timestamp sequences  $T_R$ . Their timestamp correlation,

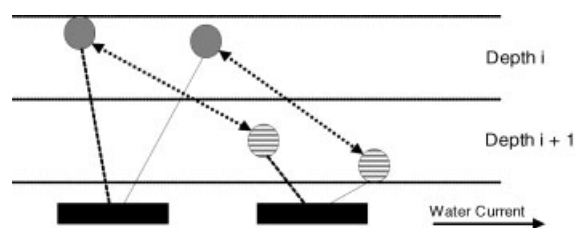


Fig. 5. Distance correlation between neighboring nodes.

$\rho_{S-R} \in [-1.0, 1.0]$ , is defined as their linear relationship:  $\rho_{S-R} = \text{Cov}(T_S, T_R) / (\sigma_{T_S} \bullet \sigma_{T_R})$ , where  $\text{Cov}(T_S, T_R)$  is the covariance between  $T_S$  and  $T_R$ , and  $\sigma_{T_S}, \sigma_{T_R}$  are sample variances of  $T_S$  and  $T_R$ . The higher value of  $|\rho_{S-R}|$  represents closer relationship between  $T_S$  and  $T_R$ . Obviously, a random attack from a malicious node  $M$  will generate very low  $\rho_{S-M}$  and  $\rho_{M-R}$ . On the other hand, the coordinated attacks from nodes  $M$  and  $N$  can produce high  $\rho_{M-N}$  (correlation between nodes  $M$  &  $N$ ), but very low  $\rho_{S-M}$  and  $\rho_{S-N}$ .

An observer sensor can use the above correlation test to detect malicious data from one of its neighboring sensor, i.e., one-to-one outlier detection case. However, in practice, an observer sensor can receive a few timestamps from multiple neighbors. Because the distances and acoustic propagation delays between the observer sensor and each of its neighbor sensors follow a certain statistical distribution, we may want to quickly find out a suspicious receiving timestamp (i.e., an outlier) from those timestamp values received from multiple neighbors. The  $T$ -test has been verified to be effective in handling such a one-to-many outlier detection case [19]. It is for comparing the means of two small populations to compensate the possible deviation from real mean  $\mu$  and population standard deviation  $\sigma$ . It rejects null hypothesis if the calculated  $t$  value is (1) higher than  $t_{\alpha}(v)$  for upper-tailed hypothesis; or (2) lower than  $-t_{\alpha}(v)$  for lower-tailed hypothesis; or (3)  $|t|$  is higher than  $t_{\alpha/2}(v)$  for two-tailed hypothesis. Here  $\alpha$  is the significance level and  $v$  is the degree of freedom.

**Definition 2.  $t$ -test:** Suppose that an observer collects multiple timestamp values (we use  $\{T_i\}$  to denote such a set of collections) from its neighbors within a window of period during which we can have  $W$  collections. Let us also assume that all timestamp values fit in a normal distribution. The  $t$ -test equation can be denoted as follows (where  $\mu$  is the sample mean,  $S$  is the sample deviation, and  $n$  is the sample size):

$$t = \frac{(\bar{T}_i - \mu)}{(S/\sqrt{n})} \quad (5)$$

For the convenience of statistical test, we need to segment the continuous timestamp samples into different windows with size of  $W$ . To reflect the impact of history data on data analysis, we keep part of data samples overlapped between two sliding windows. The following definition provides the calculation methods of window parameters.

**Definition 3. Timestamp Sliding Window:** Suppose that the underwater sensors use the sampling rate of  $F$  samples per unit time (thus data granularity is  $1/F$  time units per sample); the period of each window is  $P$  time units ( $P$  is also called precision granularity); Denote  $H$  as the overlapping factor or history weight (that is, how history is weighted in the window). Then we can calculate the window size  $W$  and actual window sliding distance  $D$  as follows:  $W = F \bullet P$ ;  $D = F \bullet P \bullet (1 - H)$ .

The procedure for the Insider Attack prevention scheme can be briefly stated as follows: (1) For a legal sensor whose identify has been verified before, i.e., an observer, it obtains the timestamp readings from its multiple neighbors. It first ignores the outlier timestamps using  $t$ -test; (2) For each neighbor's timestamp, the observer calculates the correlation coefficient. It collects a window of coefficients for outlier statistics; (3) For each window of data, it compares each coefficient to a coefficient threshold (discussed later). If the coefficient is below the threshold, it is an outlier value. The **outlier percentage** is the abnormal data percentage in one window.

**Determine the coefficient threshold  $\xi$ :** It could be a constant based on empirical data; the lower  $\xi$  yields lower false negative but may increase false positive and communication overheads. A better way to set the threshold is based on the confidence interval of coefficients in a long term  $\rho_{S-R}$ . We need to use Fisher  $Z$ -transform to normalize  $\rho_{S-R}$  as follows:

$$Z_{S-R} = 0.5 \bullet \log \left[ \frac{(1 + \rho_{S-R})}{(1 - \rho_{S-R})} \right] \quad (6)$$

This transformed correlation  $Z_{S-R}$  is normally distributed with variance  $1/(W-3)$  [19]. Hence the 95% confidence interval is:  $Z_{S-R} \pm \frac{Z_{0.025}}{\sqrt{W-3}} = Z_{S-R} \pm \frac{1.96}{\sqrt{W-3}} = (Z_{\text{Lower}}, Z_{\text{Upper}})$ . (1) If the outlier percentage is consistently (such as after 10 windows) higher than an alert threshold, the corresponding neighboring node could be a suspicious node that generates insider attacks. The value of alert threshold in our simulation is 75%. However, the value of alert threshold depends on security level requirements; if higher security level is required, a higher alert threshold can be used; (2) adjust window size based on the following algorithm until the min/max window size is reached; (3) go back to (1).

**Window size adjustment algorithm:** Let  $\Delta\rho$  represents the change in correlation coefficient from  $\rho[t]$  to

$\rho[t + 1]$ , where  $t$  represents time. Let  $\Delta O$  represent the change in outlier percentage from  $O[t]$  to  $O[t + 1]$ ; and  $\Delta w$  represents the change in sliding window size from  $W[t]$  to  $W[t + 1]$ . Thus, for an appropriate value of  $W$ , we define the following adjustment algorithm: Define the window ratio  $\gamma$  as follows:  $\gamma = \frac{\Delta W}{W} = \frac{\Delta \rho \bullet \Delta O}{2}$ , where  $-1.0 \leq \gamma \leq 1.0$ , and  $W_{\text{Lower}} \leq W \leq W_{\text{Upper}}$ . Then the new window size should be:  $W(t + 1) = (1 + \gamma) \bullet W(t)$ . From the above algorithm, we can see that  $W$  increases only if both  $\Delta \rho$

binomial, etc., can be used to represent the reputation. Here we use the beta distribution due to its flexibility and simplicity as well as its strong foundations in the theory of statistics [20,21].

**Definition 4.  $\beta$ -distribution:** *The beta-family of probability density functions (pdf) is a continuous family of functions indexed by two parameters  $\alpha$  and  $\beta$ . The beta distribution  $P(x | \alpha, \beta)$  can be expressed using the gamma function  $\Gamma$ :*

$$\text{Beta}(\alpha, \beta) = P(x|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \bullet \Gamma(\beta)} \bullet x^{\alpha-1} (1 - x)^{\beta-1}, \begin{cases} 0 \leq x \leq 1, \alpha > 0, \beta > 0 \\ x \neq 0, \text{ when } \alpha < 1 \\ x \neq 1, \text{ when } \beta < 1 \end{cases} \quad (7)$$

and  $\Delta O$  are either positive or negative and decreases when their signs differ. The above correlation test or t-test can detect outlier data from neighboring nodes from time to time. However, identifying a neighboring node as a malicious one is a difficult task since a node may temporarily function abnormally or just because the acoustic signal is not strong enough due to fading effects. Moreover the water currents can change sensors' location largely in different times, which also causes the large change of propagation delays between any two neighboring nodes. Hence, in the following Step 2 security procedure, we propose to use a long-term reputation of any node observed from all other nodes to find out whether a node is an insider attacker or not.

### 5. Statistical Trust Model

To make a decision on whether a neighbor is an insider attacker or not needs long-term behavior observations. In this section we will propose Step 2 Security procedure that can accurately identify the enemy nodes from a quantitative measurement, which is called **Reputation, or statistical trust** in this paper. Suppose that an observer underwater sensor  $\wp$ , wants to measure the trust level of one of its neighboring sensors,  $\mathfrak{J}$ , in terms of the receiving timestamp packets in different windows. We denote the reputation of node  $\mathfrak{J}$  maintained by node  $\wp$  as **Rup**( $\mathfrak{J} \rightarrow \wp$ ), which is defined as a **probabilistic distribution** to enable the sensor to have full freedom and not get constrained by some discrete levels of reputation ( $\pm 1, 0$ ) as used in eBay, Yahoo auctions. It can be used to statistically determine whether node  $\mathfrak{J}$  is cooperative or not. Several distributions such as beta, Gaussian, Poisson,

The probability expectation value of the  $\beta$ -distribution is [20]:  $E(x) = \alpha/(\alpha + \beta)$ . For any process with two very different outcomes,  $\{\Theta, \bar{\Theta}\}$ , let  $v$  be the observed occurrence times of outcome  $\Theta$  and let  $\bar{v}$  be the observed occurrence times of outcome  $\bar{\Theta}$ , then the pdf of observing outcome  $\Theta$  in the future can be expressed as a function of past observations by setting  $\alpha$  and  $\beta$  as follows:  $\alpha = v + 1$ , where  $v \geq 0$ ,  $\beta = \bar{v} + 1$ , where  $\bar{v} \geq 0$ .

**Definition 5. Sensor Reputation function:** *Based on the above characteristics of  $\beta$ -distribution, we define  $v$  to be the number of timestamp t-test windows that have outlier percentage **larger** than a predefined threshold (see Section 4), and  $\bar{v}$  to be the number of such windows with outlier percentage **smaller** than the threshold. Assuming that  $v$  and  $\bar{v}$  are all maintained in the observer sensor  $\wp$ , which receives timestamp packets from a neighbor sensor  $\mathfrak{J}$ , then **Rup**( $\mathfrak{J} \rightarrow \wp$ ), can be represented by  $\beta$ -distribution as:*

$$\begin{aligned} \text{Rup}(\mathfrak{J} \rightarrow \wp) &= P(x|\alpha, \beta) = \text{Beta}(\alpha, \beta) \\ &= \frac{\Gamma(\nu + \bar{\nu} + 2)}{\Gamma(\nu + 1) \bullet \Gamma(\bar{\nu} + 1)} \bullet x^\nu (1 - x)^{\bar{\nu}}, \\ &\text{where } 0 \leq x \leq 1, 0 \leq \nu, 0 \leq \bar{\nu} \end{aligned} \quad (8)$$

Example: For a reputation distribution, Beta(8,2), (i.e., so far 7 windows of timestamps have normal outlier percentage; and 1 window has unsatisfied outlier percentage), its pdf can be shown as in Figure 6. This curve expresses the probability that the timestamp window has higher or lower than a threshold during future observations. F(x|8,2) has an expecta-

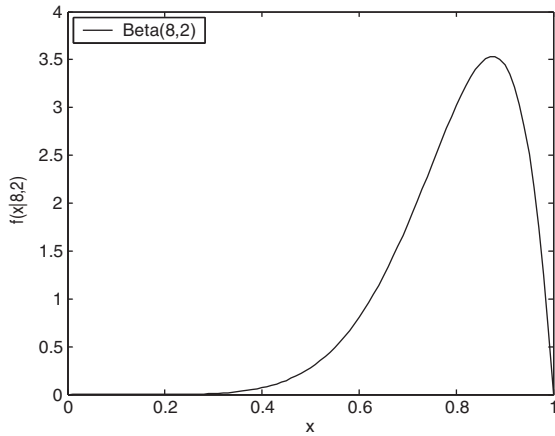


Fig. 6. Beta (8,2).

tion of  $E(x)=0.8$ , which means that the relative frequency of good windows in the future will most likely be 0.8.

**Definition 6. Sensor Trust:** Equation (8) is a statistical function and cannot be used to directly measure the trust level of a sensor. To obtain a single-value trust metric for any sensor  $\mathfrak{S}$ , we define  $\mathfrak{S}$ 's trust (to  $\mathfrak{F}$ ) as the expectation of the reputation function:  $\mathbf{Trust}(\mathfrak{S} \rightarrow \mathfrak{F}) = E(\mathbf{Rup}) = (\nu + 1)/(\nu + \bar{\nu} + 2)$ . The flexibility of using the  $\beta$ -distribution is to measure continuous, updated 'Trust' through the following simple approach:

**Theorem 1.** Let us assume that a node  $\mathfrak{F}$  has already built a reputation metric of its neighbor  $\mathfrak{S}$ ,  $\mathbf{Rup}(\mathfrak{S} \rightarrow \mathfrak{F})$  by time  $T$ . From time  $T$  to  $T + Now$ , node  $\mathfrak{F}$  receives  $\zeta + \xi$  more windows, out of which the  $\zeta$  windows have unsatisfied outlier percentage (i.e., higher than a threshold) and  $\xi$  windows have satisfied outlier percentage. The reputation and trust value can be updated based on the following relationship:  $\nu_{\text{NEW}} = \nu + \zeta$ , and  $\bar{\nu}_{\text{NEW}} = \bar{\nu} + \xi$ .

*Proof.* The basic Baye's theorem is used to calculate the probability of a belief given an observation.  $P(\text{Belief/Observation}) = P(\text{Observation/Belief}) * P(\text{Belief})/\text{Normalization}$ , where  $P(\cdot)$  is determined by Equation (8). In our context, 'belief' analogously represents the reputation of a sensor and the observation refers to the maintenance of windows of timestamps within a certain observing period. If  $\mathfrak{F}$  observes  $\mathfrak{S}$ 's timestamp packets and finds out  $\nu$  unsatisfied windows (i.e., outlier percentage is higher than a threshold) and  $\bar{\nu}$  good windows, node  $\mathfrak{F}$  can predict  $\mathfrak{S}$ 's behavior (friendly or abnormally),  $\Omega$ , for

the next event. If  $\mathfrak{F}$  does not have any prior timestamp windows,  $\Omega$  is uniformly distributed over the measurement space,  $(0,1)$ . Thus:  $P(\Omega) = \text{uniform}(0, 1) = \text{Beta}(1, 1)$ . Using the binary rating model, we can model the prior interactions using a binomial distribution and then the posterior distribution of  $\Omega$  can be calculated as [20]:  $P(\Omega) = \frac{\text{Binary}(\nu+\bar{\nu}) \bullet \text{Beta}(1,1)}{\text{Normalization}} = \text{Beta}(\nu + 1, \bar{\nu} + 1)$ .

Please note that the problem in Theorem 1 is analogous to the above one, although now the prior distribution is represented by a beta and not a uniform distribution. The reputation  $\mathbf{Rup}(\mathfrak{S} \rightarrow \mathfrak{F})$ , can be updated as:

$$\begin{aligned} \mathbf{Rup}(\mathfrak{S} \rightarrow \mathfrak{F}) &= \frac{\text{Binary}(\zeta + \xi, \zeta) \bullet \text{Beta}(\nu + 1, \bar{\nu} + 1)}{\text{Normalization}} \\ &= \text{Beta}(\nu + \zeta + 1, \bar{\nu} + \xi + 1) \end{aligned} \tag{9}$$

That is, the reputation update is equivalent to just updating the value of two parameters  $\nu$  and  $\bar{\nu}$  as follows:  $\nu_{\text{NEW}} = \nu + \zeta$ , and  $\bar{\nu}_{\text{NEW}} = \bar{\nu} + \xi'$ . The updated single-value Trust will then be:

$$\mathbf{Trust}(\mathfrak{S} \rightarrow \mathfrak{F}) = E(\mathbf{Rup}_{\text{Refresh}}) = \frac{\nu + \zeta + 1}{\nu + \bar{\nu} + \zeta + \xi + 2} \tag{10}$$

**Aging reputation:** The recently obtained timestamp information should be given higher weight since it is more important than the past timestamps. In this research, we incorporate exponential averaging in the following way: (where  $\text{Weight}_H$  represents the importance of past data and with range  $(0,1)$ ):

$$\begin{cases} \nu_{\text{NEW}} = \text{Weight}_H \bullet \nu + \zeta \\ \bar{\nu}_{\text{NEW}} = \text{Weight}_H \bullet \bar{\nu} + \xi \end{cases}$$

Please note that the setup of  $\text{Weight}_H$  can also ensure that nodes cooperate all the times. A malicious node can possibly cooperate in the beginning but abuse the system thereafter using the reputation acquired initially. An appropriate choice of  $\text{Weight}_H$  will ensure that the reputation information and the trust value becomes stale, which also means that a node should cooperate continuously to maintain a good reputation.

## 6. Experimental Results

We have investigated our proposed UAN security performance through OMNET++ [22] and Matlab



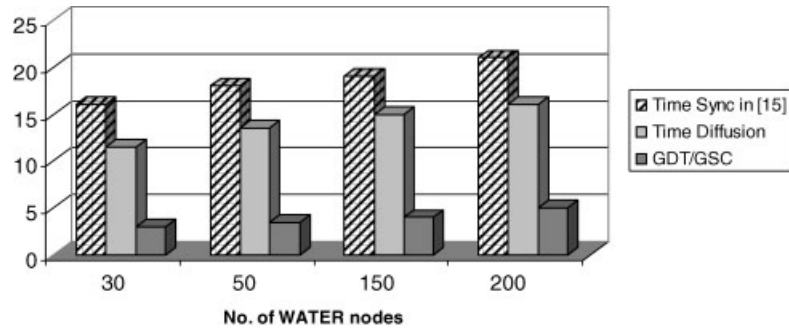


Fig. 9. Time synchronization overheads.

than 70% of total energy [1]. As shown in Figure 9, the time diffusion scheme consumes most energy, which may come from its global cluster organization and its slow Allan variance evaluation. Our scheme consumes the least energy because we have closely integrated the GDT/LSC procedure with the vertical tree topology.

We next evaluate the performance of our scheme in overcoming the effects of high acoustic propagation delay. Especially, we compare its performance to the time synchronization scheme in Reference [2]. For this, we measure the absolute clock offset error (which is defined as the difference between the global time and the corrected local time in a node) as a function of propagation delays (which will be different for different communication distance). In our tests, we intentionally make the deepest node further and further from the root and then collect the absolute clock offset errors as shown in Figure 10. It is not surprising to see that time diffusion [13] has the highest clock offset errors when used in UAN due to no consideration of

compensating clock skew. Even our GDT/LSC has slightly higher offset errors than [2]. Overall, it has comparable performance while saving more energy than [2].

### 6.1. On Connectivity Establishment for 'Horizontal' Synchronization

In Subsection 3.2, we discussed the importance of 'orphan sensor to non-orphan sensor' connectivity establishment (at the same water depth) for the UAN horizontal time synchronization. We exported the horizontal topology discovery results (including absolute positions and acoustic links) to Matlab for better observations. Figure 11 shows the results of our 3-D connectivity establishment algorithm, where the hollow nodes are orphans and the solid ones are non-orphan sensors that have parents. Thus an orphan sensor can search a closest non-orphan sensor to achieve horizontal clock adjustment. Our reputation-based security scheme could be used for common sender-receiver based or receiver-receiver based synchronization approaches. It is especially suitable for receiver-receiver based schemes since it can accurately estimate the trust level of neighboring sensors. In our experiments, we have implemented the cluster-based time diffusion scheme [13].

In our experiments, we add a random delay just before the MAC layer sends out timing packets, which makes the point-to-point delay not follow normal distribution. Thus the sending/receiving timestamps between two neighboring sensors will lose good correlation. We set up a sensor to collect 5 PH samples per minute, i.e.,  $F=5$ ; with window period  $P=30$  min and  $H=0.50$ . Based on Definition 3,  $W$  is 150 and  $D$  is 75. The acoustic propagation delays (after normalization) between any two neighboring nodes, which can be easily collected from OMNET

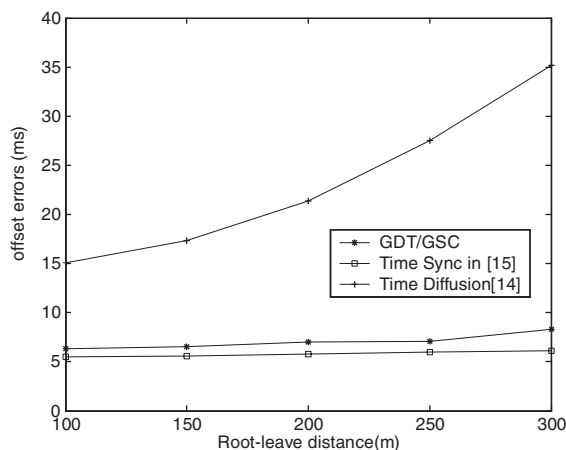


Fig. 10. Synchronization accuracy versus acoustic delays.

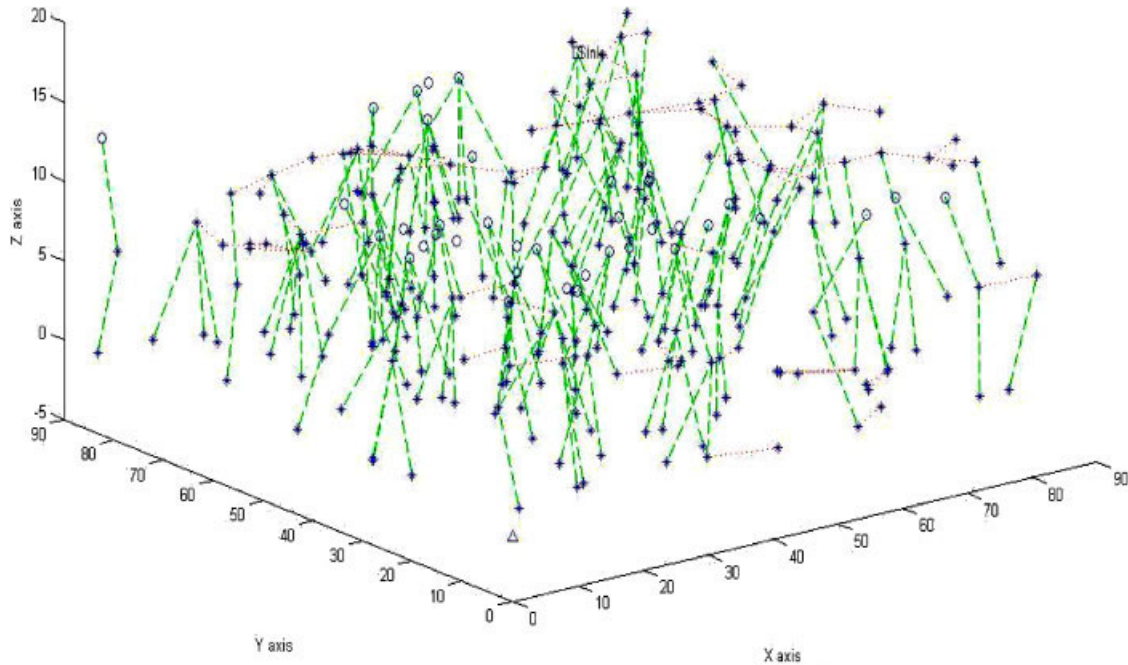


Fig. 11. Horizontal connectivity establishment.

link layer statistics, have been verified to be an approximately normal distribution based on our normality tests  $P-P$  plot (see Figure 12), which means the timestamps between any neighbors should have good correlation without insider attacks.

### 6.2. On Statistical Tests

Because our reputation-based scheme depends on statistical tests, we have repeated each simulation

scenario for 75 times for each of the following tests. Each time we have used different water sensor energy storage, mobility speeds and acoustic noise levels.

**(A) Acoustic links without insider attacks:** In acoustic links without insider attacks, we expect that the timestamp information between the two nodes shows close correlation. In our large-scale simulations (with 200 nodes, 10 water depth levels), we have collected the timestamp values for three nodes A, B and D that are located at depth 5, 6 and 7, in that order (Figure 13). Their timestamp data correlation results are shown in Figure 14. The nodes A and B and B and D have higher correlation coefficients than nodes A and D, which show that neighboring nodes have closer correlation than longer distance nodes.

**(B) Acoustic links with insider attacks: Attack model:** We assume that one sensor has become ‘malicious’ (due to being captured by enemies and then all of its keying materials have been disclosed). It uses ‘flooding attack’, i.e., it keeps sending wrong timestamp messages to all of its neighbors. The close correlation is lost if we set up node B (in Figure 13) as an insider attacker. With smaller window size, the correlation coefficient in each window has higher fluctuations and can thus ‘zoom in’ the effects of insider attacks (see Figure 15).

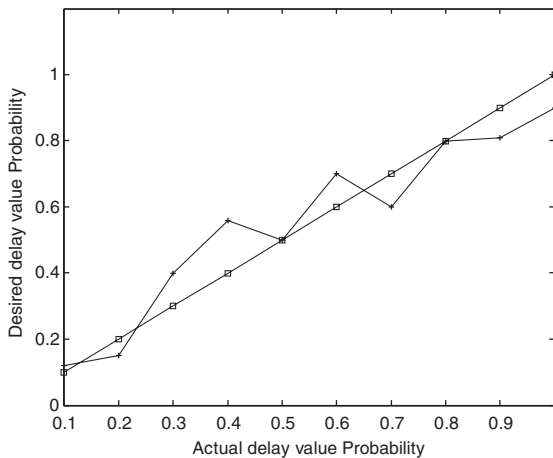


Fig. 12. Normal  $P-P$  plot of acoustic delays.

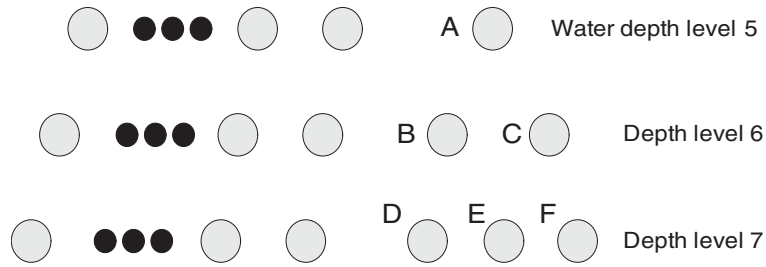


Fig. 13. Simulation topology on security.

Node ID	A	B	D
A	1	-	-
B	0.94	1	-
D	0.83	0.93	1
Node ID	A	B	D

Fig. 14. Timestamp data correlation results.

**(C) Reputation test:** To test the efficiency of our statistical trust model, within a large-scale UAN (with 250 nodes), for an observer ♀, we set up one neighbor as ‘honest’ and another neighbor as ‘malicious’ (see Figure 16). We use *t*-test (see Section 4) to observe the outlier percentage in different timestamp windows. A window with high outlier percentage (> 50%) is identified as a non-cooperation event and lower than 50% is regarded as a cooperation event. We then

calculate the statistical trusts based on reputation distributions of two neighbors (as shown in Figure 17). After the collection of only dozens of windows of timestamp data, we can well indicate the malicious node (see ‘neighbor 2’ in Figure 17). We can also see that the trust values have sharp change in the first 10–30 windows and then slowly approach a stable value, which shows that our proposed statistical trust model has good convergence characteristics. Figure 18 clearly shows the impact of using ‘aging reputation’ in attack identification. We have given the recent ‘cooperation’ events higher weight compared to the history data. If a neighbor cooperates in the beginning and then becomes malicious later on, Figure 18 shows that aging reputation model punishes it. Thus we see that the node’s trust values decrease at a faster speed for ‘aging reputation’ case than in ‘no aging reputation’ case (Figure 18).

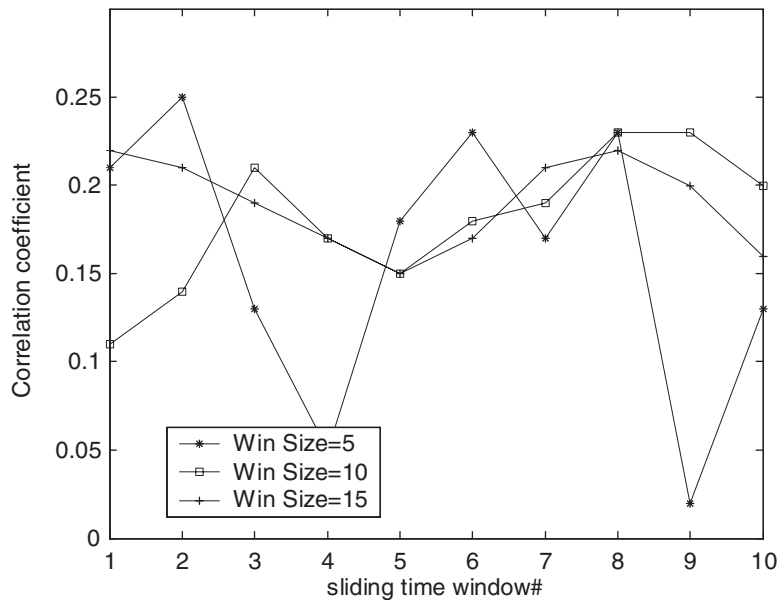


Fig. 15. Correlation coefficient for insider attack case (between node A and B and node B is an attacker).

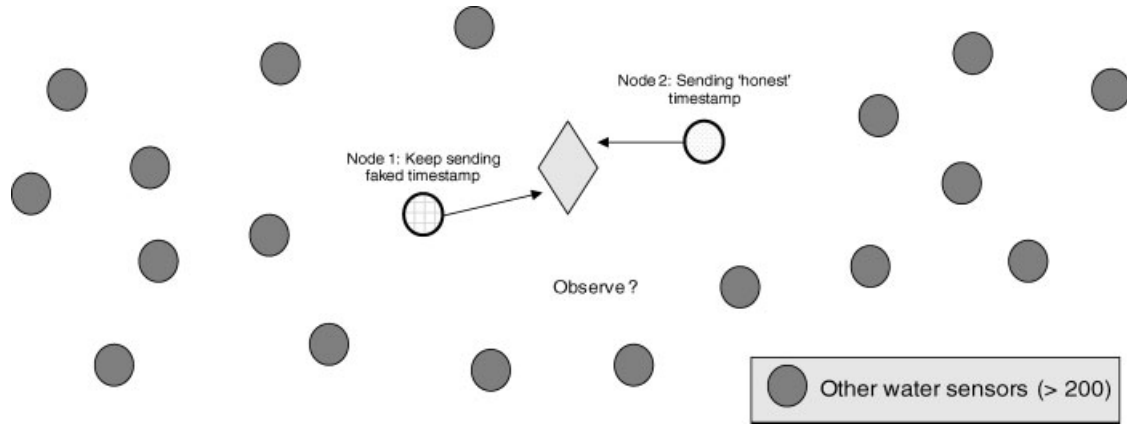


Fig. 16. Reputation test nodes.

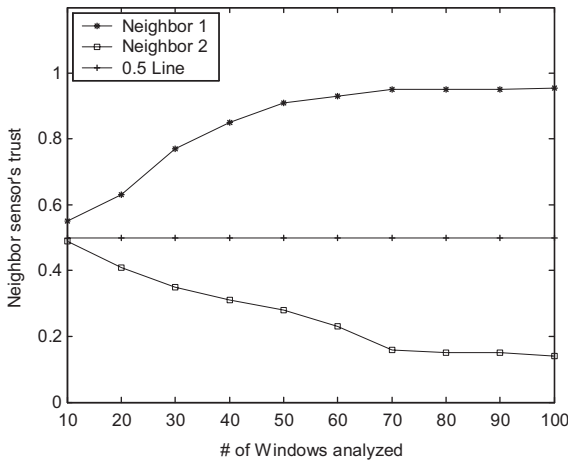


Fig. 17. Reputation test results.

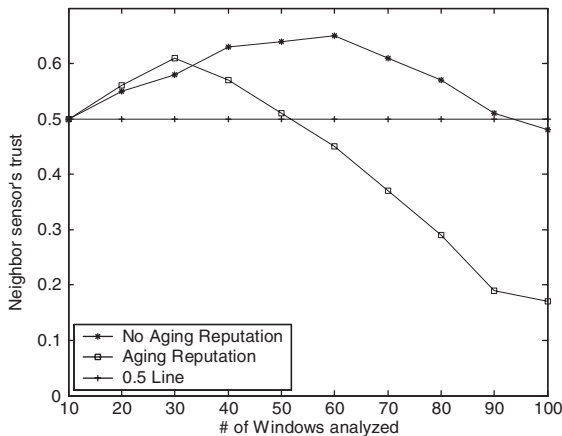


Fig. 18. Aging reputation test.

### 7. Conclusions

In this paper, we have described our research results on building an insider-attack-resistant, delay-tolerant vertical/horizontal synchronization mechanism in the UAN platform, which has essential differences from terrestrial radio sensor networks due to its highly variable, long propagation delay and mobility nature. In the vertical direction, our light-weight time synchronization mechanism can achieve satisfactory timestamp accuracy. We have also addressed the horizontal synchronization issue that is different from vertical synchronization. Moreover, we have proposed a two-step security model for time synchronization services in underwater acoustic sensor networks: Step 1 security uses correlation test to detect outlier timestamp data; and Step 2 security uses statistical reputation and trust model to identify nodes generating insider attacks, which is different from external attacks due to the complete keying material disclosure in the insider attack events. Our experimental analysis has shown the efficiency of our window-based outlier detection protocol and reputation-based attacker identification. In future, we shall study the wormhole attacks and corresponding countermeasures in underwater acoustic networks.

### Acknowledgments

This work was supported in part by the Cisco URP Grant, RIT Faculty Development grant, and the US National Science Foundation (NSF) under grants of DUE- 0511098, CNS-0716455, and CNS-0716211.

## References

1. Akyildiz IF, Pompili D, Melodia T. Challenges for efficient communication in underwater acoustic sensor networks. *ACM Sigbed Review* 2004; **1**(2): 3–8.
2. Syed A, Heidemann J. Time synchronization for high latency acoustic networks. *Technical Report ISI-TR-2005-602*, USC/Information Sciences Institute, April, 2005.
3. Hu F, Tilghman P, Malkawi Y, Xiao Y. A prototype underwater acoustic sensor network platform with topology-aware MAC scheme. *International Journal of Sensor Networks (IJSNet)* 2007; **2**(5/6): 386–398.
4. Gibson JH, Xie GG, Xiao Y, Chen H. Analyzing the performance of multi-hop underwater acoustic sensor networks. In *Proceedings of IEEE/OES Oceans 07 Aberdeen Conference*.
5. Gibson J, Xie GG, Xiao Y. Performance limits of fair-access in sensor networks with linear and selected grid topologies. In *Proceedings of IEEE GLOBECOM 2007*.
6. Sundararaman B, Buy U, Kshemkalyani AD. Clock synchronization in wireless sensor networks: a survey. *Ad-Hoc Networks* 2005; **3**(3): 281–323.
7. Manzo M, Roosta T, Sastry S. Time synchronization attacks in sensor networks. In *Proceedings of The Third ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, Alexandria, VA, USA, 2005.
8. Ganeriwal S, Capkun S, Han C, Srivastava MB. Secure time synchronization service for sensor networks. In *Proceedings of the 4th ACM Workshop on Wireless Security*, September, 2005.
9. Song H, Zhu S, Cao G. Attack-resilient time synchronization for wireless sensor networks. *Technical Report*, Department of Computer Science & Engineering, The Pennsylvania State University, 2005.
10. Zhou D. Security issues in ad hoc networks. In *The Handbook of Ad Hoc Wireless Networks*. CRC Press: New York, 2003; 569–582.
11. Fober D, Orlarey Y, Letz S. Clock skew compensation over a high latency network. In *Proceedings of the ICMC*. ICMA, 2002.
12. Ganeriwal S, Kumar R, Srivastava M. Timing sync protocol for sensor networks. *ACM SenSys*, Los Angeles, November 2003.
13. Su W, Akyildiz IF. Time-diffusion synchronization protocol for wireless sensor networks. *IEEE/ACM Transactions on Networking* 2005; **13**(2): 384–397.
14. AODV (Ad hoc On Demand Distance Vector (AODV) routing algorithm): see <http://moment.cs.ucsb.edu/AODV/>.
15. Spohn MA, Garcia-Luna-Aceves JJ. Enhanced dominant pruning applied to the route discovery process of on-demand routing protocols. In *Proceedings of IEEE IC3N 03*, 2003.
16. Wu J, Cardei M, Dai F, Yang S. Extended dominating set and its applications in ad hoc networks using cooperative communication. *IEEE Transactions on Parallel and Distributed Systems* 2006; **17**(8): 851–864.
17. Allan D. Time and frequency (time-domain) characterization, estimation, and prediction of precision clocks and oscillators. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 1987; **34**(6): 647–654.
18. Urick R. *Principles of Underwater Sound*. McGraw-Hill Book Company: New York, 1991.
19. Hogg RV, Tanis EA. *Probability and Statistical Inference* (7th edn). Prentice Hall: Englewood Cliffs, NJ, 2005.
20. Ganeriwal S, Srivastava MB. Reputation-based framework for high integrity sensor networks. *ACM Security for Ad-hoc and Sensor Networks (SASN)*, 2004.
21. Resnick P, Zeckhauser R. Reputation Systems: Facilitating Trust in Internet Interactions, 2000. on: <http://www.si.umich.edu/~presnick/papers/cacm00/reputations.pdf>.
22. OMNET++ community site: <http://www.omnetpp.org>.

## Authors' Biographies



**Fei Hu** is an Associate Professor in the Department of Computer Engineering at RIT (Rochester Institute of Technology), New York, USA. His research interests are wireless networks, wireless security and their applications in Bio-Medicine. His research has been supported by NSF, Cisco, Sprint, and other sources. He obtained his first Ph.D. degree at Shanghai Tongji University, China in 1999, and second Ph.D. degree at Clarkson University, USA in 2002, all in the field of Computer Engineering. He has published over 100 journal/conference papers and book (chapters). He is also the Editor for over five international journals.

**Yamin Malkawi** is a Graduate student in Computer Engineering Department at RIT. His research interests are wireless sensor networks and next-generation internet.



**Sunil Kumar** received B.E. (Electronics Engineering) degree from S.V. National Institute of Technology, Surat (India), in 1988 and the M.E. (Electronics & Control Engineering) and Ph.D. (Electrical and Electronics Engineering) degrees from the Birla Institute of Technology and Science (BITS), Pilani (India) in 1993 and 1997, respectively. He also served as a Lecturer in the Electrical and Electronics Engineering Department at BITS, Pilani (India) from January 1993 to July 1997. From August 1997 to August 2002, he was a Postdoctoral Researcher and Adjunct Faculty in Signal and Image Processing Institute, Integrated Media Systems Center and Electrical Engineering–Systems Department at University of Southern California, Los Angeles. From July 2000 to July 2002, he was also a Consultant in industry on JPEG2000 and MPEG4-based projects and participated in JPEG2000 standardization activities. From August 2002 to July 2006, he was an Assistant Professor in the Electrical and Computer Engineering Department at Clarkson University, Potsdam, NY. Since August 2006, he has been an Associate Professor and Thomas G. Pine Faculty Fellow in the Electrical and Computer Engineering Department at San Diego State University, San Diego, CA. His research interests include (i) QoS-aware Protocols for Multimedia Traffic in Wireless Cellular, Adhoc, Sensor, and Cognitive Networks, (ii) Error Resilient Multimedia (Image, Video and 3-D graphics) Compression techniques, including MPEG-4, H.264/AVC and JPEG2000, (iii) Image Processing techniques with applications in biomedical and fingerprint images, and (iv) ANN-GA based Machine Learning techniques for bioactivity prediction and data mining of drug molecules (Chem-Bioinformatics). He has published more than 75 research articles in international journals and conferences, and one book. His research has been funded by NSF, DOE, NYSERDA, CITeR, Cisco, and Sprint.



**Yang Xiao** worked in industry as a MAC (Medium Access Control) architect involving the IEEE 802.11 standard enhancement work before he joined Department of Computer Science at The University of Memphis in 2002. He is currently with Department of Computer Science at The University of Alabama. He was a voting member of IEEE 802.11 Working Group from 2001

to 2004. He is an IEEE Senior Member. He is a member of American Telemedicine Association. He currently serves as Editor-in-Chief for *International Journal of Security and Networks (IJSN)*, *International Journal of Sensor Networks (IJSNet)*, and *International Journal of Telemedicine and*

*Applications (IJTA)*. He serves as a referee/reviewer for many funding agencies, as well as a panelist for NSF and a member of Canada Foundation for Innovation (CFI)'s Telecommunications expert committee. He serves as TPC for more than 100 conferences such as INFOCOM, ICDCS, MOBIHOC, ICC, GLOBECOM, WCNC, etc. He serves as an Associate Editor or on editorial boards for several journals (such as *IEEE Transactions on Vehicular Technology*, etc.). His research areas are security, telemedicine, sensor networks, and wireless networks. He has published more than 200 papers in major journals (more than 50 in various IEEE journals/magazines), refereed conference proceedings, book chapters related to these research areas. Dr. Xiao's research has been supported by the US National Science Foundation (NSF).