Localization with witnesses

Arun Saha and Mart Molle
Department of Computer Science and Engineering
University of California, Riverside
Riverside, CA, 92521, USA
{saha, mart}@cs.ucr.edu

Abstract

Localization protocols enable an entity (called the verifier) to determine the physical location of another entity (called the prover), even if the prover maliciously advertises a false location or tries to corrupt the verifier's time measurements by time-shifting its responses. Unfortunately, the correctness of such protocols is critically dependent on the verifier's ability to make high-resolution time measurements and on the prover's ability and trustworthiness to send its response by the mandated time. To address these problems, we propose the idea of incorporating passive witnesses into the localization protocol. All witnesses monitor the same bilateral packet exchange between the prover and lead verifier and later report their respective inter-packet time measurements to the lead verifier for further processing. We show how the extra information provided by the witnesses can eliminate the threat of response-time shifting by a malicious prover. We also pose the question, how can we combine multiple localization observations to a single localization estimate? While analyzing that, we observe that the localization estimate is sensitive to the relative position of the prover among the verifiers.

Index Terms

Ad-hoc network, Sensor network, Localization, Time-Difference-of-Arrival.

I. Introduction: A case for Location Authentication

The membership to an adhoc network or a sensor network is generally dynamic. It is interesting to explore the situation when the membership criteria to such wireless, self organizing networks is based on proximity and relative distances among the devices.

We consider a wireless network where requests for membership roles are granted if the requesting device is "sufficiently" close to the existing network, i.e. existing members. The semantics of "sufficient" closeness is decided by the network. A straightforward approach to address this requirement is to make the requesting device mention its position while sending the membership request. In a perfect world where all devices are going to be truthful, there is no problem. However, there might be some incentives to be part of such a wireless network which might tempt a malicious device to claim any arbitrary position of its choice.

Example 1 Some wireless sensors are spread on a environmental experimental testbed to keep the temperature, humidity in control. A rival organization who want to steal the experimental procedure might place some sensors outside the testbed yet inside the transmission range and attempt to join the sensor network.

Example 2 There is a query to an environmental sensor network, whichever sensor is closest to a particular position is asked to report the temperature. A malicious sensor who is not actually the closest to the target can claim its position such that it appears closest and subsequently responds to the query with incorrect data. This might either raise a false alarm or subvert a true alarm.

Example 3 All laptops which are inside a building are assumed to be carried by employees or their guests; they are allowed to join the network and access the Internet if requested. Here, people outside the building carrying laptops might be tempted to claim an inside position, thereby gain access to the Internet, and unauthorizedly use the bandwidth.

In these ways, there might be different kind of undue advantages to be gained if a malicious device can join a position/proximity based network when it actually is not located in a place to do that. This is the motivation to correctly determine the location of such a requesting device. There are two minor variations. The requesting device can claim a position which the existing members then verify. Or, the existing members can determine the location of their own. Whichever it is, the existing members do not trust the claimer but collaborate among themselves to decide.

II. Background

The first notion is who is performing the localization activity. In one approach, known as self-localization, the mobile entity collect information from the neighborhood and determine its own location. One such example is Global Positioning System (GPS), where the GPS receiver device receives/collects information from the GPS satellites and determines its own location. In the other approach, the neighbors of the mobile entity, whose location is of interest, collect information, and combine them

A. Saha is currently with Cisco Systems

to determine its location. One example of this approach is E-911 calls from cellphones where the location of the caller is determined by the cellphone base stations.

In the last decade, a number of localization systems were proposed, mainly based on infrared, ultrasound, radio signal and ultra wide band. In our work, we refer to localization in the context of mobile ad-hoc network or sensor network, here the communication between the entities are wireless radio communication.

The localization algorithms are mainly based on (i) received signal strength, (ii) angle of signal arrival (AoA), (iii) or time of signal arrival (ToA) measurements, or their combinations. One important variation of ToA system is the Time Difference of Arrival (TDoA) system.

Another broad categorization of localization approaches is based on whether the distance between the entities are measured or not. Time based systems finally convert the time measurements to distance measurements. The localization systems which are based on the distances between the entities are known as range-based systems, e.g. [1]. Others are called range-free or range-independent, e.g. [2]. Another important characteristic is whether the localization is infrastructure based or ad-hoc. The GPS self-localization or the E-911 localization are based on infrastructure.

The entities participating in the localization can use traditional omnidirectional antenna or smart directional antenna. Smart antennas can transmit and receive energy in one direction as opposed to disseminate in all directions. However, we feel that they will defeat the simplicity of the system since we are targeting the entities in mobile ad-hoc or wireless sensor networks using standard networking protocols.

In ToA based solutions, each verifier executes a distance bounding protocol and determines an upper bound of the distance to the prover. Accurate timing measurements are required to obtain the round trip signal propagation time (and hence distance) between the verifier and the prover. The basic distance bounding is proposed by Brands and Chaum [3] where the verifier sends a single-bit challenge and the prover responds with a single-bit response "immediately after" receiving the challenge. Such challenge-responses are carried out for multiple rounds and the verifier measures the round-trip time at each round. The verifier then computes the upper-bound of the distance based on the maximum of the round-trip times. The above concept is applied and further extended by Capkun and Hubaux [4], Capkun, Buttyan and Hubaux [5], Hancke and Kuhn [6], Reid *et al.* [7] etc. These approaches have multiple implementation constraints as mentioned in [8], [9].

There are significant differences between ToA based solutions and TDoA based solutions. In ToA, the disadvantages are: (i) multiple (at least three) verifiers have to undergo separate challenge-response dialog with the prover, and (ii) the prover's delay between receiving the challenge and sending the response called response delay (note that it happens for each challenge-response, and they are independent!) affects the final localization result. However, in ToA the verifiers need not be time-synchronized, but in TDoA that is a requirement.

The model of localization in this work is the following. There is a set of entities, called verifiers, who want to localize another entity whom we call prover. The verifiers and the prover use omnidirectional radio-frequency communication. The localization system is range based without any infrastructure. Also note that, in this paper, we are mainly focusing on localization concepts ignoring the cryptographic security of the protocol messages; in reality these concepts need to be tightly coupled as in [4], [8], [9].

III. The wireless secure localization problem

The problem can be formalized as the following. The formation of a wireless network A is based on locality constraints. In most cases, it is a single hop network i.e. every member can receive any transmission by other members. The n members of the network are designated as $A_1, A_2, ...A_n$. There is new node V which claims to be in the vicinity of the network and wants to join. The existing members collaborate to determine the location of the requesting node and decide on the request.

A. Assumptions

The goal of the verifiers is to localize the prover using the existing standard network hardware and protocols.

1) Trust model: The existing members of the wireless network A_i $(i = 1...n)$ mutually trust each other and co-operate, they are known as verifiers. The new entity, known as the prover, is completely untrusted. The prover can be located anywhere with respect to the verifiers, and can take any amount of time in responding to messages.

2) Mobility: The wireless entities can be possibly mobile. However, we assume that during the execution of the localization protocol, the group of wireless entities are relatively in rest. For example, a group of wireless entities might actually be a fleet of cars in a highway and all of them are moving at a constant speed.

3) Co-ordinate System: We assume that there is a local co-ordinate system. The wireless entities included in the network know their location in that coordinate system. The entities may be possibly equipped with GPS receivers but that is neither necessary nor sufficient to solve the problem.

4) Transmission Range: A good number of wireless entities already included in the network must be able to receive the transmission from the requesting entity. The entities which will be receiving transmission from the requesting entity are the verifiers. The verifiers will take part in the execution of the protocol.

IV. Using witnesses to prevent distance fraud attacks

A. Motivational example of a one-dimensional network

Figure 1(a) shows a typical timed-echo challenge-response dialog between the verifier V and prover U using the space-time representation described in [10]. For simplicity, we will temporarily assume that the network is a one-dimensional broadcast network, such as a coaxial cable shared Ethernet segment. In addition, we summarize each message by a single arrow, representing the start-frame delimiter at the beginning of its transmission, rather than a shaded region covering its entire transmission time. In this example, V starts a timer at point (1) from which the challenge spreads in both directions away from V as time advances down the page. After propagation delay α_{VU}, the leading edge of the challenge arrives at point (2), where U can start to receive it. After formulating a suitable reply, U begins to transmit its response at point (3). After a further propagation delay α_{UV}, the leading edge of U's response arrives at point (4), where V stops its timer and continues to receive the remainder of the message. Unfortunately, even with perfect timing accuracy, V still cannot determine its distance to U by measuring the inter-packet time between points (1) and (4), τ_V, because U can vary the inter-packet time between points (2) and (3), τ_U. Indeed, we say that a malicious prover is launching a *distance fraud attack* if it secretly changes τ_U from the value expected to force V into calculating an incorrect value for $\alpha_{VU} \leq \alpha_{VX} \equiv \frac{1}{2}\tau_V$.

In Figures 1(b)-(d) we show how a passive witness W to the same challenge-response dialog between U and V can help the verifier avoid distance fraud attacks. (Recall that the physical layer is assumed to be an omni-directional broadcast channel, so the same challenge transmission that left V at point (1) also reaches W at point (5), and the same response transmission that left U at point (3) also reaches W at point (6).) First, in Fig. 1(b) we assume that witness W_1 is located on the opposite side of V from prover U. In this case, the inter-packet time measurements τ_V made by V between points (1)-(4) and τ_{W_1} made by W_1 between points (5)-(6) will be identical, which shows only that the rays from U to W_1 and from V to W_1 must be parallel, and hence U and V must be on opposite sides of V. In this case the witness cannot help the verifier to localize the prover.

Next, in Fig. 1(c) we assume W_2 is located somewhere between V and U. In this case, the triplets $(1)-X-(4)$, $(5)-X-(6)$, and $(2)-X-(3)$ all form similar triangles, from which we obtain

$$\tau_{W_2} \equiv \tau_V - 2\alpha_{VW_2} \tag{1}$$

and hence that U must be further from V than W_2 in the same direction. Finally, in Fig. 1(d) we assume that W_3 is located beyond U on the same side of V. This time, the inter-packet time measurement τ_{W_3} made by W_3 between points (5)-(6) must be exactly the same as τ_U ***whether or not*** U ***tries to spoof*** V***'s measurement of*** τ_V. In these two cases (Fig. 1(c) and Fig. 1(d)), V can easily determine the location of U by substituting all known values into Eq. (1) by solving for the unknown propagation delay:

$$\alpha_{VU} = \frac{\tau_V - \tau_{W_i}}{2} \tag{2}$$

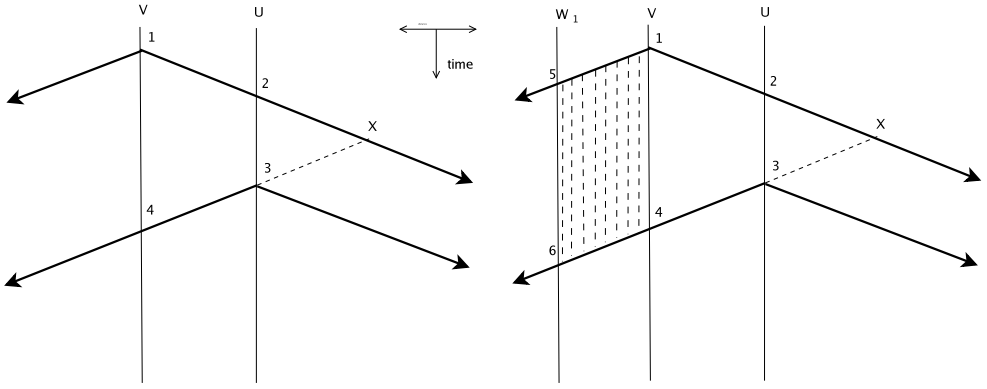
B. General approach for wireless broadcast networks

The simple one-dimensional example described above can be generalized to create a novel and *secure* localization technique for a group of trusted co-operating nodes, say $\{A\}$, that communicate over a wireless, planar[1] broadcast network. Our approach uses a single bilateral query-response dialog between one member of the group, V called the *lead verifier*, and an untrusted non-cooperating prover, U. One of the verifiers is (s)elected as lead-verifier. Other group members who are within transmission range of both the lead-verifier and the prover, serve as passive witnesses. The technique does not depend on distance-bounds or RTT from verifier to prover. The basic concept is similar to the Time-Difference-Of-Arrival (TDoA) techniques as summarized in [8]; however we apply it differently for localizing a possibly-fraudulent prover.

Similar to other localization protocols (e.g., [11]),, there is an initial untimed setup phase in which U approaches the group for verification of its claimed location so it can participate in various location-dependent activities. The group chooses a lead-verifier, V, and a set of witnesses, and then V instructs U to prepare for the bilateral message exchange phase. This second phase consists of only two messages: the *challenge* sent by the lead-verifier V, and a *response* sent by prover U. Notice that the witnesses are completely passive during this phase: they receive all messages while sending none of their own.

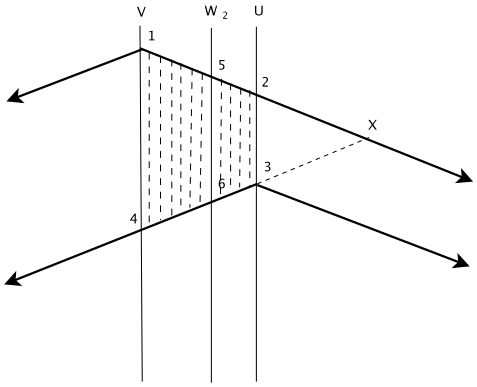
Once the prover receives the (real) challenge, it computes the response and transmits it. The inter-packet time from the instant that the prover starts receiving the challenge until the instant that it starts transmitting its response is called the **response delay** and denoted τ_U. During this second phase, all active group members monitor the packet exchange between V and U and record the interpacket-time from the instant they either start receiving (if a witness) or transmitting (if the lead verifier) the challenge, until the instant they start receiving the response. Therefore, at the end of the challenge-response dialog, the active group members report their respective inter-packet time measurements: τ_V for lead-verifier V and τ_{W_i} for the ith witness.

[1]The generalization to three dimensional geometry is straightforward but tedious, and will not be considered further.

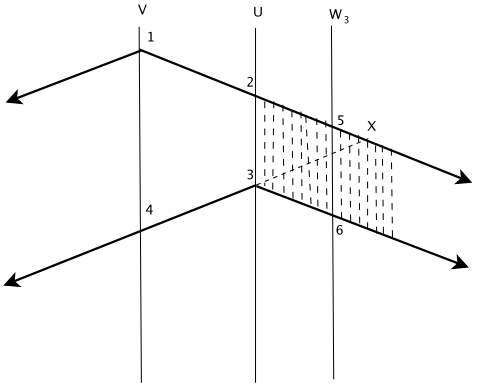


(a) The verifier and the prover

(b) The verifier in the middle.



(c) The witness in the middle.



(d) The prover in the middle.

Fig. 1. Witness can help localization in one-dimension. 1(a) shows the verifier and the prover. Subsequent figures show three possible relative locations of the witness with respect to the verifier and the prover. (Figures are not to scale.)

Let the distance and signal propagation delay between two entities X and Y be denoted by D_{XY} and α_{XY} respectively. If the signal propagation speed in the medium is v, then $\alpha_{XY} = D_{XY}/v$. Now for witness W_1 we have,

$$\begin{aligned} \tau_{W_1} &= \alpha_{VU} + \tau_U + \alpha_{UW_1} - \alpha_{VW_1} \\ &= D_{VU}/v + \tau_U + D_{UW_1}/v - D_{VW_1}/v \end{aligned} \tag{3}$$

For another witness $W_2 (W_2 \in A \setminus W_1)$, we have,

$$\tau_{W_2} = D_{VU}/v + \tau_U + D_{UW_2}/v - D_{VW_2}/v \tag{4}$$

Subtracting Eq. (4) from Eq. (3),

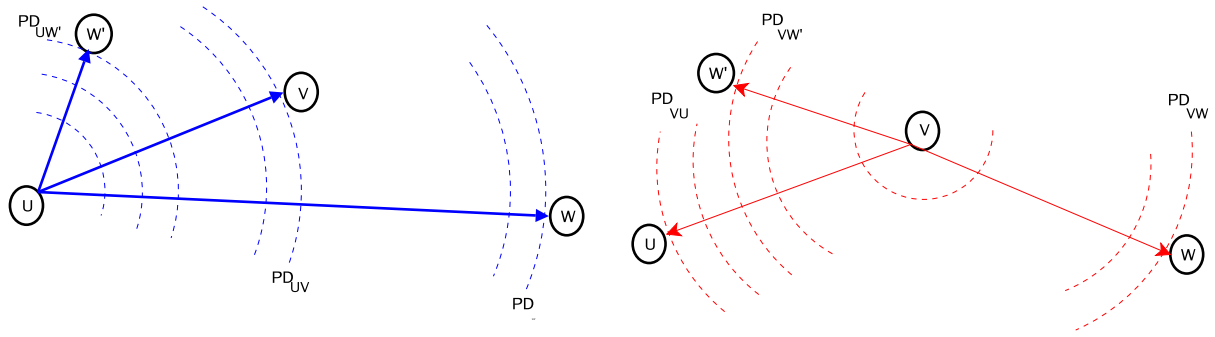
$$\tau_{W_1} - \tau_{W_2} = \frac{1}{v}\{(D_{UW_1} - D_{UW_2}) - (D_{VW_1} - D_{VW_2})\}$$

Transposing,

$$D_{UW_1} - D_{UW_2} = v(\tau_{W_1} - \tau_{W_2}) + (D_{VW_1} - D_{VW_2}) \tag{5}$$

Let us assume that the entities are located in a two dimensional plane. Suppose the positions of U, V, W_1 and W_2 be $U(x_U, y_U)$, $V(x_V, y_V)$, $W_1(x_{W_1}, y_{W_1})$ and $W_2(x_{W_2}, y_{W_2})$ respectively. The distances D_{VW_1} and D_{VW_2} are known,

$$\begin{aligned} D_{VW_1} &= \sqrt{\{(x_{W_1} - x_V)^2 + (y_{W_1} - y_V)^2\}} \text{ and} \\ D_{VW_2} &= \sqrt{\{(x_{W_2} - x_V)^2 + (y_{W_2} - y_V)^2\}} \end{aligned}$$



(a) The verifier transmitting the real challenge.

(b) The prover transmitting the response.

Fig. 2. Different stages of the localization protocol. V is the lead-verifier, U is the prover, and W, W' are witnesses. Due to the nature of the broadcast medium, all messages are heard by the witnesses as well. Signal propagation delay from entity i to entity j is denoted by a_{ij}.

Substituting similarly in Eq. (5) we obtain,

$$\begin{aligned} & \sqrt{\{(x_U - x_{W_1})^2 + (y_U - y_{W_1})^2\}} - \sqrt{\{(x_U - x_{W_2})^2 + (y_U - y_{W_2})^2\}} \\ = \; & v(\tau_{W_1} - \tau_{W_2}) \\ + \; & (\sqrt{\{(x_{W_1} - x_V)^2 + (y_{W_1} - y_V)^2\}} - \sqrt{\{(x_{W_2} - x_V)^2 + (y_{W_2} - y_V)^2\}}) \end{aligned} \tag{6}$$

Since V, W_1 and W_2 are mutually-trusted and co-operating, they can exchange their location and the independently measured τ values, i.e. τ_V, τ_{W_1}, τ_{W_2} respectively. So all the terms in Eq. (6), except (x_U, y_U), are known. Hence Eq. (6) is the locus of the unknown position (x_U, y_U). In particular, the locus is one of the two arcs of a hyperbola.

We observed above that *any* two verifiers can find out the locus of the prover. Similarly, another independent locus of the prover can be formed by combining the time interval data of a different pair of verifiers. The two loci, i.e. the two equations, can be solved to find out the location of the prover. Thus any three verifiers, including or excluding the lead-verifier, can localize the prover. Other things like distance, round trip time and response delay can be easily derived.

Positions of two witnesses can be initialized during the network startup. Moreover, once a prover's position becomes known, it can possibly be used as a witness in future.

Note that the Eq. (6) does not depend on the response delay τ_U. Hence the final solution is independent of response delay at the prover. Thus this technique is resistant to Distance Fraud attack where the prover can intelligently enlarge and reduce distances to fool a set of three verifiers and spoof a different location [1], [4].

C. Results:

The above analysis leads us to the following results:

- Any verifier-pair can form the locus of the prover.
- Any verifier-triplet can localize the prover.
- The location found by a verifier-triplet is independent of the response delay (τ_U) at the prover.

V. Accurate measurement of the Time Interval τ_{W_i}

There are two challenges with regard to the measurement of the Time Interval τ_{W_i}:

- The time interval measurements should be fine grained. Radio signal travels in vacuum at a speed of $c \approx 3 * 10^8 m/s$. So, the context of localization application and the tolerances are important. For example, when we target localization within a room or building, we should target error tolerance in meters, not *hundreds* of meters. Reversing the argument, if we want the distance estimation within error tolerance of meters, then the maximum error tolerance in timing measurement must be in the order of nanoseconds.
- The time interval measurements should be free from clock skew. The time intervals measured by all the verifiers should be based on a single clock as there might be clock skew among the verifiers' local clocks. Of course that single clock should meet the required high precision requirements.

In this section we describe how the above two challenges can be addressed.

A. Features of wireless communication

There is one feature in wireless communication which is distinct from the wired world. When there is a one-hop connection between two wired hosts, then whatever they transmit in the medium is known only by those two hosts and no other host.[2] On the other hand, since the wireless medium is basically broadcast in nature, any unicast message transmission can be "heard" not only by the intended receiver but also by all other entities in the sender's transmission range. In particular, if the channel is not reserved by some other neighbor, a wireless entity *has* to receive all packets in the medium at least to determine whether the packet is destined for it. This is what we mean by "heard".

B. Fine grained time interval measurement cannot be done in software

For correct localization in range-based systems, the true signal propagation time is required; the time interval measurement should exclude time spent in all other activities. If the desired time interval is measured in the application layer of the verifier/witness, then it includes the time of the message traversal through the verifier/witness protocol stack. Also, if the response is generated at the application layer of the prover then the message traversal time through the prover protocol stack is also included in the measured interval. Traversal time through the protocol stack includes time for passing the message among different layers of hardware and software. This additional delay is unpredictable and of much higher order of magnitude than the propagation delay. Similar arguments hold true if the time interval is measured in operating system software. The reader is requested to refer [8, Sec. 3.1.9] for additional details and sample figures regarding this.

C. Fine grained time interval measurement with help from hardware

Our approach is to perform the measurement in a place inside the device which is very close to the device to medium interface. Using current technology we can achieve it by moving the measurement task all the way down to bottom of the protocol stack, where we utilize the first hardware component adjacent to the medium-dependent signaling interface, the PHY.

There are two main functions performed by the Physical Coding Sublayer (PCS) inside the physical layer (PHY): (i) generation of continuous code-groups to be transmitted on the channel, and (ii) processing of the code-groups received. In addition to the code-groups corresponding to the frame data, the transceivers use some control code-groups. One such control code-group is Start Frame Delimiter (SFD), which indicates the arrival of a new frame.

We assume that the verifiers, i.e. the PHY of the verifier, are able to detect the reception time of a specific marking code-group when they receive a frame. By specific marking code-group, we mean something particular like Start Frame Delimiter (SFD) code-group of the frame.

The lead-verifier can measure the time interval τ_V as follows: it starts a timer when the SFD of challenge frame is transmitted, and stops it when the SFD of response frame is received.

Due to the inherent property of the wireless communications mentioned above, the challenge frame transmitted by the lead-verifier will be "heard" by all the other verifiers i.e. witnesses. Hence the witnesses can measure the time interval as the elapsed time between the following two events: (i) reception of the SFD-CG of the challenge frame from the lead-verifier, and (ii) reception of the SFD-CG of the response frame from the prover.

Let us now examine the solution in little more detail. Once a witness hear the dummy Challenge before the real Challenge from the lead-verifier, the witness understands that a challenge-response dialog is about to begin. The witness then keep its transceiver in ready-to-receive state until finally the response arrives from the prover.

However, the PHY is not capable of interpreting the contents of a received frame. So, the PHY by itself cannot know when the challenge frame is going to come, and correspondingly start the timer on receipt of the marked code-group. But, with instructions from the higher layers, the PHY can do so. Therefore, once the dummy Challenge is received, the higher layer instructs the PHY that now is the time that the PHY should start the timer on the receipt of the marked code-group of the next arriving frame. Once instructed, the PHY will start the timer on the receipt of the marked code-group of the next arriving frame and stop the timer on receipt of the same marked code-group in the subsequent frame.

D. Measuring time interval using common clock

When an entity receives a frame, it changes state from IDLE to RECEIVING, and goes through a synchronization process. The synchronization process is responsible for determining whether the underlying receive channel is ready for operation. After bit synchronization, the receiver knows the bit transmission rate of the sender. In other words, the receiver acquires the clock rate of the sender's PHY. Once that is done, the receiver can setup a local timer with frequency equal to the sending PHY's transmitter clock.

Thus when an witness "hears" or receives the challenge frame, it sets up a local timer with the frequency of the lead-verifier's PHY transmitter. This timer is used measure the time interval τ_{W_i}. The interval is measured in terms of the number of clock ticks of that timer. One clock tick interval if that timer is equal to the symbol time of the lead-verifier's PHY. All the witnesses use this method. This way all the measured time intervals are in terms of a single clock and free from clock skew errors.

[2]However, generally two hosts seldom have direct one-hop connection, the connection goes through some network devices e.g. repeater, switch or router. That is, direct single hop connections are seen between a host and a network device or between two network devices.

E. Discussion

The PHY will report the time interval to the higher layers only if it is requested, and that reporting will be much later after the receipt of the response frame. Also note that, there will always be maximum of one such time interval measurement result in the PHY. If the higher layer instruct the PHY to make another time interval measurement, then the PHY will overwrite its previous measured value. This is because the main localization algorithm is carried in the application layer, the PHY needs to make measurement once for each execution of the algorithm.

We note that due to the multipath nature of the wireless channel, frame transmissions will experience multi-path delay spread. A code-group radiated using an omnidirectional antenna, will take multiple paths (as a consequence of reflections from various objects) to arrive at the receiver. In other words, the receiver will receive multiple copies of the same signal, each of which may have a different amplitude, phase and delay. One received symbol will interfere with other copies of its own. Due to this fact, the exact reception time of a code-group is difficult to characterize. One possible approximation is to consider the first copy since the line-of-sight path will frequently be the quickest.

VI. Computational issues

A. Measurement errors

Like any other measurement, the time intervals τ_{W_i} noted by the verifiers are subject to error – the measured time intervals might be little too high or little too low. Such measurement noise will affect the locus of the prover and subsequently its location. In that case, the solution points of two different verifier-triplets will not be exactly the same. But, if the measurement errors are not large, we can expect all those solutions points to be scattered around the actual location (unknown to the verifiers) of the prover. There might be some extraneous solution points however.

B. An over-determined system

Since any three verifiers can collaborate to localize the prover, then the next natural question is the following. If there are more than three verifiers, which three of them will be chosen to localize the prover? For each arbitrary choice of a verifier-triplets, we can determine a possible position for the prover. If there are n verifiers, then the number of verifier-triplets can be formed is $N = \binom{n}{3}$. We will get one (two in some cases) solution point from each set. In total, there will be approximately N solution points, i.e. N possible locations for the prover. Such a system is often referred to as an "over-specified" or "over-determined" system, a potential drawback of using an over-determined system relates to the fact that hyperbolic localization algorithms can calculate more than one mathematically valid position [12]. The situation is like a fallacy and counter-intuitive from the statistical point of view. When we had less information it was easier to conclude, when we have more information it is difficult!

C. Combining multiple solution points

Now, what is needed is some method to use *all* these solution points to make a *single* final estimate about the location of the prover V. The most naive choice, the mean of all the solution points as (mean of all x-coordinates, mean of all y-coordinates), is not good because of the fact that arithmetic mean is highly affected by the outliers. However, the median of the solution points might be good. (Zhang *et al.* [13] takes the median of K distance-estimates.) There, one option is to output the point (median of all x-coordinates, median of all y-coordinates) as the final estimate. Another option is to find the two-dimensional median of the points, i.e. the central-most point among all. One simple way to do that is to find the 2D-median as described below.

2D median: Construct a convex polygon with a subset of the solution points, such that all the remaining solution points which are not the vertexes of the polygon are inside the polygon. Then discard the solution points included in the polygon, and repeat the process with the remaining solution points. In this way of repeatedly peeling-off outer points, the central-most solution points (maximum of three) can be found. One of these, or their mean, can be the final estimate.

Another approach of combining the multiple solution points is the following: Imagine all the solution points obtained from different sets of verifiers as different measurements of the same signal and use them to make a final estimate. Kalman Filtering is one possible way to do that.

D. Kalman Filtering

Kalman filtering is an optimal, recursive, discrete data processing algorithm. It addresses the general problem of trying to estimate the state of a discrete-time controlled process that is governed by linear stochastic difference equation [14]. The algorithm predicts the state ahead, makes a measurement, then combine the prediction and measurement such that the error covariance is minimized. Again it makes prediction for next stage and so on.

	Prover P1(1, 2)		Prover P2(4, 15)	
	No Error	With Error	No Error	With Error
ToA	Fig.3(a)	Fig.3(a)	Fig.4(a)	Fig.4(c)
TDoA	Fig.3(b)	Fig.3(b)	Fig.4(b)	Fig.4(d)

TABLE I
EXPERIMENT SUMMARY.

E. Kalman Filtering to combine multiple solution points

We experimented with Kalman Filtering to estimate the prover's location. First, we obtained all the possible solution points by pairwise solving all the hyperbola equations. Then we passed the solution points one by one through the Kalman Filter. After sufficient number of steps, the estimate converges.

However, as we observed in our simulation experiments, the order in which different solution points are considered by the filtering algorithm significantly affect the final estimate. The same set of solution points processed in different order produces different final estimate. Thus there is a need to find out a way to order the different solution points such that the final estimate is as close to the actual location as possible.

We believe that the orientation of the verifier-triplet, and the location of the prover relative to that orientation is of importance. Recall from the earlier example of one dimensional network, if the witness is on the opposite side of the prover (Fig. 1(b)), then the verifier cannot localize the prover. Some solution points and their associated verifier-triplet are more significant than others. More significant solution points should be treated earlier than the less significant ones. We propose the following heuristics:

1) If the solution point lies within the triangle formed by the verifier-triplet, then that solution point is more significant than if the solution point lies out of the triangle.
2) If the verifier-triplet is almost collinear, then the solution obtained from them will be poorer than from the verifier-triplet which constitute a well-formed triangle.
3) A solution point which is closer to all locus curves is expected to be nearer to the actual location compared to a solution point which is not. To achieve that, the normal distance from each solution point to all the locus curves are found and added up. Then the solution points are ordered in decreasing order of aggregate distances to be considered by the Kalman filter

In the following section we analyze the first heuristic.

F. Sensitivity of prover location w.r.t. verifier-triplet

If the solution point lies outside the triangle formed by the verifier-triplet, then it is very sensitive to measurement error. In such cases, a little measurement error displaces the probable solution points by a (relatively) large amount. This is shown in the following example.

Three verifiers, A_1(-5, 0), A_2(0, 5) and A_3(8, 0) are trying to localize a prover. In our experiment, we consider:

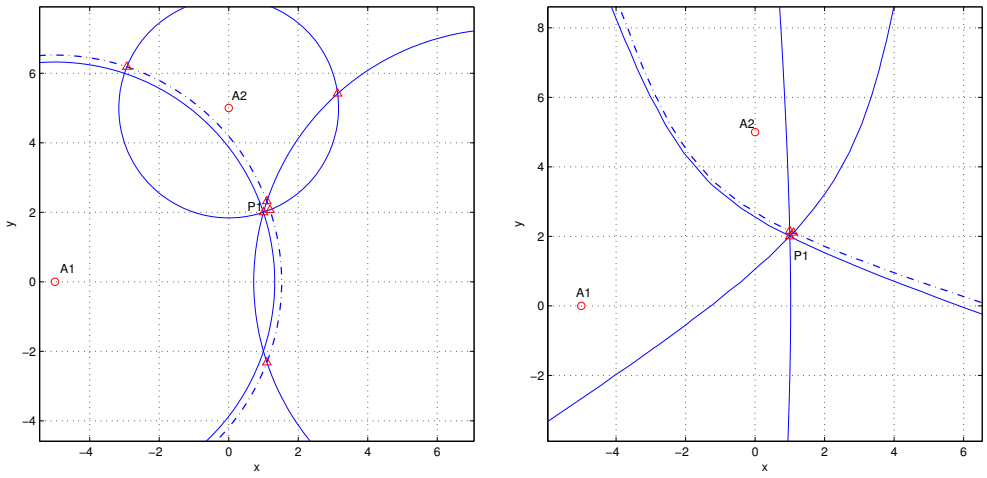
- Two locations of the prover V: (i) inside the triangle $\Delta A_1A_2A_3$ as P_1(1, 2), (ii) and outside the triangle as P_2(4, 15).
- Two methods of localization: (i) ToA [3] (intersection of circles) method, (ii) and TDoA (intersection of hyperbolas) method.
- Two cases of measured data: (i) with no measurement error, (ii) and with measurement error (distance error for ToA, difference of distance error in TDoA).

Table I connects the cases described above to the diagrams shown below. Curves obtained when there is no measurement error are shown with solid lines, the dot-dashed lines show the curves obtained with measurement error. Measurement error was injected by increasing the 'distance' (in ToA) or the 'difference-in-distance' (in TDoA) by a small amount.

The solid-line circles in Fig.3(a) show how the three verifiers localize P1 when there is no measurement error. But when there is measurement error (here, in A_1's measurement), the three circles do not intersect at any single point, however there are two intersection points which are *very close* to the actual solution point. In Fig.4(a), we see how the three verifiers localize P2 when there is no measurement error. However, Fig.4(c) shows that when measurement errors are present, the derived intersection points are *not very close* to the actual location of P2. The following simple logic indicates that the intersection points moves more when they are outside the triangle as opposed to when they are inside the triangle. The Fig. 3 and the Fig. 4 have the same zoom level. In Fig. 3 where the prover is inside the triangle, the intersection points generated from erroneous measurement are almost superimposed on the actual location point. In Fig. 4 where the prover is outside the triangle, the intersection points generated from erroneous measurement are distinctly visible from the actual location point.

The same observations are repeated for the TDoA method as shown in Figs 3(b), 4(b), and 4(d).

[3]The solution proposed in this document (§IV) uses the TDoA method. Still we consider the ToA method in this example since that is another widely used method.



(a) ToA method

(b) TDoA method

Fig. 3. Sensitivity of errors when the prover is at location (1,2) which is inside the triangle formed by the verifier-triplet.

In a different experiment, the measurement noise with both positive and negative values are considered. When there is no measurement error, verifiers A_1 and A_2 generate the hyperbola H_{12} (see Fig. 5). With a small positive offset added to the difference-in-distances, they generate the hyperbola $H_{12}p$ shown with dashed line. Similarly, with a small negative offset added to the difference-in-distances, they generate the hyperbola $H_{12}m$ shown with dotted line. H_{13} and H_{23} are the hyperbolas generated from (A_1, A_3) and (A_2, A_3) pairs respectively. The three hyperbolas — H_{13}, H_{23}, and one from $H_{12}p$ and $H_{12}m$ — gives four intersection points that form a diamond shaped patch area surrounding the actual location of the prover V. If the errors injected in the experiment is the upper bound of permissible measurement error, then that patch area denotes the possible place where the prover might actually be located. The area of the patch quantifies the uncertainty of measurement: the lesser the area, lesser is uncertainty in prover's localization. If the system of three hyperbolas yield two solution points, then there will be two patch areas; however they will surround the respective solution points (see Fig. 5(b)).

Fig. 6 shows the patch areas for different locations of the prover. The actual prover location is denoted by an asterisk inside its patch. For dual solutions, there are some patches with no asterisks inside. Note in the figure that, for prover locations closer to the verifier triangle – like (-2, 2), (-1, 2), (1, 2), (2, 2), (-2, -2), (2, -2) – the patch is almost invisible. This intuitively suggests that when the prover is actually located closer to the verifier triangle, the uncertainty in localization is lesser.

In ToA localization, where distance-enlargement attacks are possible, there is a philosophy where a verifier-triplet accepts a prover location only if the location is inside the verifier triangle. See for example the "Point in the triangle" test in [4] or the similar "Point in a polygon" test in [13]. In TDoA localization, where distance-enlargement attacks are *not* possible, the above philosophy might still hold true.

VII. Conclusions

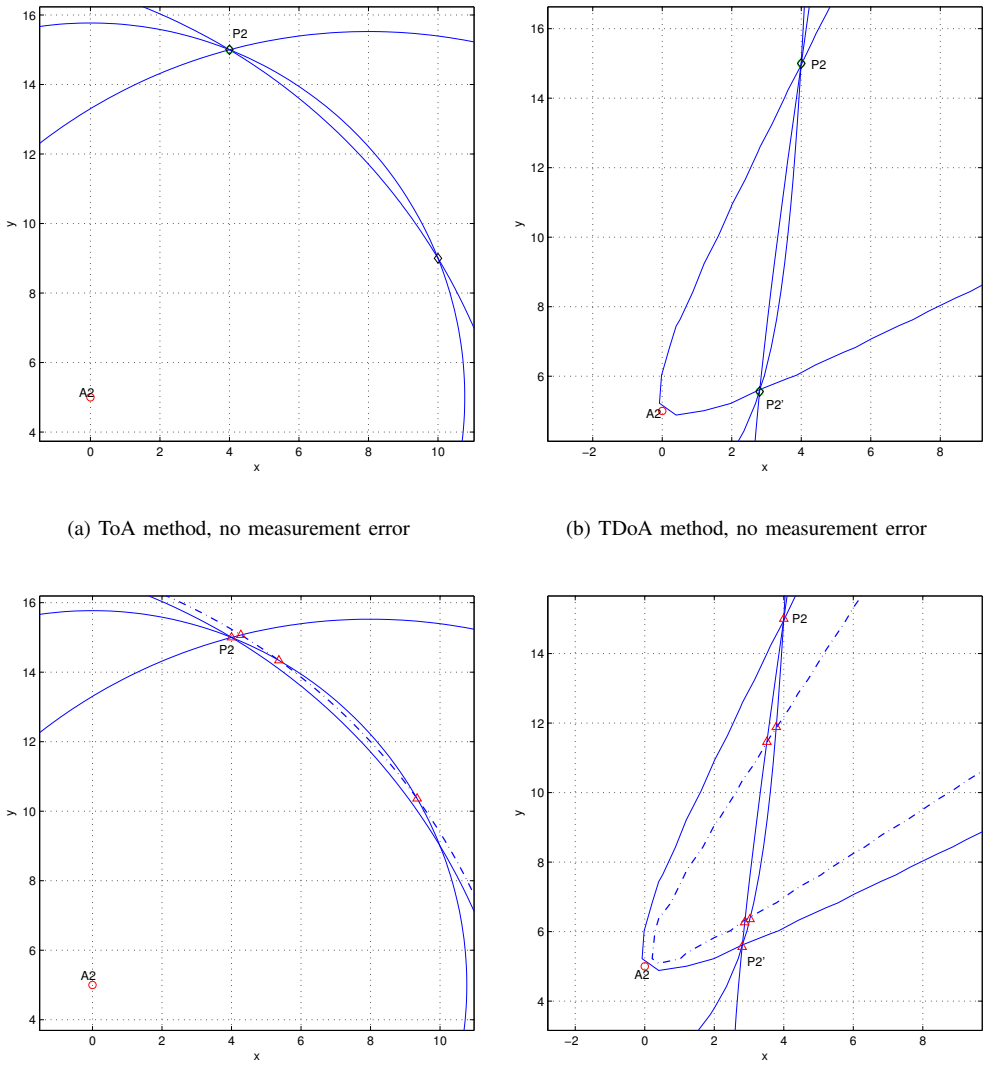
This paper analyzes some problems of localization in infrastructure-less wireless networks. We propose an approach where a passive listener can help in the localization process. Unlike some other systems where all the verifiers need to synchronize to the global clock and measure the time of occurrence of one specific event, here the verifiers need to measure a time interval between two specific events. Our solution is secure in the sense that it does not depend on the time taken by the prover to compute the response of the challenge from the verifier. We propose cross-layer co-operation between localization software and network interface hardware to measure that time interval in high precision as required for localization. Further, we introduce the problem as to how multiple erroneous observations from multiple verifiers can be combined to a single less-erroneous solution.

References

[1] S. Capkun and J.-P. Hubaux, "Secure positioning in wireless networks," *IEEE J. Select. Areas Commun.*, vol. 24, no. 2, pp. 221–232, Feb. 2006.

[2] L. Lazos and R. Poovendran, "Serloc: secure range-independent localization for wireless sensor networks," in *WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security*. New York, NY, USA: ACM Press, 2004, pp. 21–30.

[3] S. Brands and D. Chaum, "Distance-bounding protocols," in *Advances in Cryptology – EUROCRYPT ' 93*, ser. Lecture Notes in Computer Science, T. Helleseth, Ed., vol. 765, International Association for Cryptologic Research. Springer-Verlag, Berlin Germany, 1994, pp. 344–359.



(a) ToA method, no measurement error

(b) TDoA method, no measurement error

(c) ToA method, with measurement error

(d) TDoA method, with measurement error

Fig. 4. Sensitivity of errors when the prover is at location (4,15) which is outside the triangle formed by the verifier-triplet.

[4] S. Capkun and J.-P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *IEEE INFOCOMM*, vol. 3, Miami, USA, Mar. 13–17, 2005, pp. 1917–1928.

[5] S. Capkun, L. Buttyan, and J.-P. Hubaux, "SECTOR: Secure tracking of node encounters in multi-hop wireless networks," in *ACM Workshop on Security of Ad Hoc and Sensor Networks*, vol. 1, 2003.

[6] G. P. Hancke and M. G. Kuhn, "An RFID distance bounding protocol," in *IEEE/CreateNet SecureComm 2005*, Athens, Greece, Sept. 5–9, 2005.

[7] J. Reid, J. M. G. Nieto, T. Tang, and B. Senadji, "Detecting relay attacks with timing based protocols," 2006. [Online]. Available: http://eprints.qut.edu.au/archive/00003264/

[8] A. Saha, "Cross Layer Techniques to Secure Peer-to-Peer Protocols for Location, Adjacency, and Identity Verification," Ph.D. dissertation, University of California, Riverside, 2006.

[9] L. Bussard, "Trust Establishment Protocols for Communicating Devices," Ph.D. dissertation, Ecole Nationale Sup'erieure des T'el'ecommunications, France, 2004. [Online]. Available: http://www.geocities.com/laurentbussard/papers/phdThesisBussard04.pdf

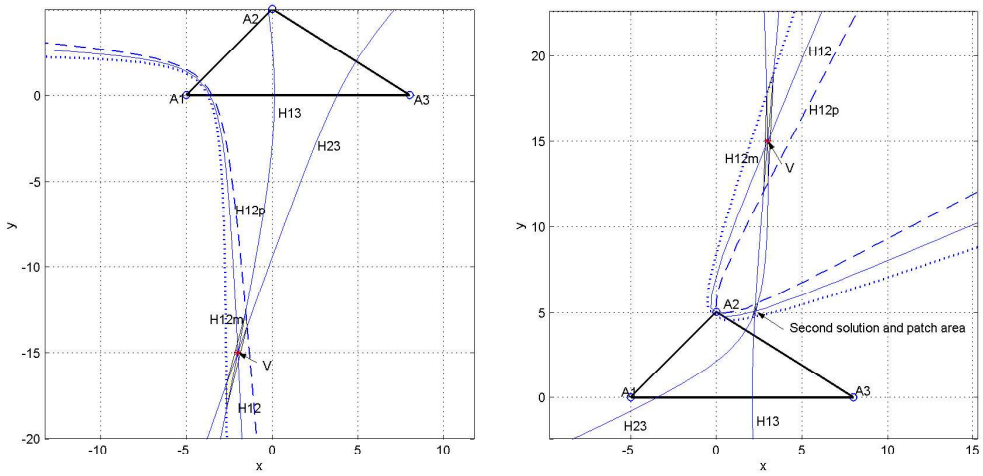
[10] M. Molle, K. Sohraby, and A. Venetsanopoulos, "Space-Time Models of Asynchronous CSMA Protocols for Local Area Networks," *IEEE J. Select. Areas Commun.*, vol. 5, no. 6, pp. 956 –968, 1987.

[11] B. R. Waters and E. W. Felten, "Secure, Private Proofs of Location," Department of Computer Science, Princeton University, Tech. Rep. TR-667-03, Jan. 2003. [Online]. Available: http://www.cs.princeton.edu/research/techreps/TR-667-03

[12] R. J. Fontana, E. Richley, and J. Barney, "Commercialization of an Ultra Wideband Precision Asset Location System," in *IEEE Conference on Wideband Systems and Technologies*, Nov. 2003.

[13] Y. Zhang, W. Liu, and D. Wu, "Secure Localization and Authentication in Ultra-Wideband Sensor Networks," *IEEE J. Select. Areas Commun.*, vol. 24, no. 4, pp. 829–835, Apr. 2006.

[14] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," University of North Carolina at Chapel Hill, Tech. Rep. TR 95-041, Apr. 2004.



(a) Prover at (-2, -15)

(b) Prover at (+3, +15)

Fig. 5. The patch areas surrounding the prover location.

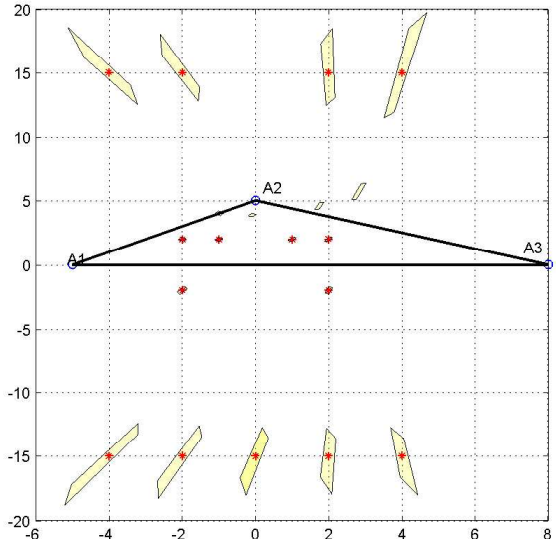


Fig. 6. Patch areas surrounding different possible prover locations.