Mobile Netw Appl (2010) 15:664–679
DOI 10.1007/s11036-010-0231-0

Timing-Based Mobile Sensor Localization in Wireless Sensor and Actor Networks

Ghalib A. Shah · Ozgur B. Akan

Published online: 9 March 2010

Abstract In this paper, localization problem in wireless sensor and actor networks (WSAN) is addressed. In WSAN, the performance of event detection and tracking highly depends on the exact location information of the events that must be reported along with the event features. Having precise location information of the sensor nodes, actors are able to execute actions more effectively in the region of detected events. In this context, the accurate localization of sensor nodes is essential with respect to the actors. Particularly, the problem becomes much more complicated when the sensor nodes as well as the anchor nodes (actors) are mobile. In order to localize the mobile sensor nodes relative to the actors, a novel Timing-based Mobile Sensor Localization (TMSL) algorithm is introduced. In TMSL, sensor nodes determine their distance from actors by using propagation time and speed of RF signal. In order to determine distance from the actors, actors actively broadcast *reference beacons* in a pattern of intervals adaptively defined according to the mobility of sensor nodes and the required level of localization accuracy. These reference beacons carry the interval numbers in which they were transmitted. The interval numbers are then used by the sensor nodes to calculate the start time of the beacons locally which is then used to determine the propagation time. TMSL does neither require nor assume any time synchronization among the sensor nodes or with the actors. Performance evaluations clearly show that TMSL is adaptive to velocity of mobile sensor and actor nodes and can be configured according to the required localization accuracy in order to avoid overhead raised due to high velocity.

Keywords timing-based localization · wireless sensor and actor networks · mobile sensor networks

This work was supported in part by the Turkish Scientific and Technical Research Council Career Award under grant #104E043 and by the Turkish National Academy of Sciences Distinguished Young Scientist Award Program (TUBA-GEBIP).

G. A. Shah
Department of Computer Engineering, National University of Science and Technology, Rawalpindi, Pakistan
e-mail: ghalib@ceme.nust.edu.pk

O. B. Akan (✉)
Next-generation Wireless Communications Laboratory, Department of Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey
e-mail: akan@eee.metu.edu.tr

1 Introduction

Wireless sensor actor networks (WSAN) represents a new computing class that consists of a large number of sensor nodes as well as, relatively, small number of actor nodes which are embedded in their operating environments. All of these nodes are often distributed over wide geographical areas or located in remote and largely inaccessible regions [1]. Sensor nodes are tiny, resource-constrained devices capable of sensing, computation and communication. On the other hand, actors are mobile, resource-rich devices capable of making decisions by themselves and executing actions on certain input received from the sensor nodes. This allows us to instrument, observe, and respond to the physical world on scales of space and time that was impossible in the past.

The emergence of this new class of networks raises many system design challenges. Since sensors are closely coupled to the physical world, they must be made aware of their geographical position to represent the area under observation. Such information is mandatory for actors to actively respond to the region of interest. A large variety of applications in WSAN ranging from military surveillance to health care system have been emerged where the location information is considered vital along with the event readings. Actor nodes require the exact location of the sources in order to identify the location where an event originates and effectively act upon the event [1]. For instance, in case of fire, sensors relay the exact origin and intensity of the fire to water sprinkler actors so that the fire can easily be extinguished before being spread uncontrollable. In health applications, location awareness facilitates doctors with the information of nearby medical equipments and personnel in a smart hospital. In mines, miners are equipped with heart beat sensors to monitor the miners health condition as well as temperature and toxic gas sensors to monitor the working conditions. In an emergency situation, an evacuating operation performed by a mobile rescue team needs exact location of the troubled miners walking around to escape.

In order to make sensor and actor networks operate effectively, a low-cost and accurate localization algorithm is imperative. Actors are usually location-aware nodes equipped with some localization hardware. However, the location of sensor nodes is unknown and is an essential requirement in WSAN. The mobility of sensor nodes, in some applications, make it a more challenging task since the nodes need to be continuously localized that incurs extra overhead. This demands an adaptive solution that provides location accuracy to certain tolerable level for reducing overhead. The size and cost factors mainly preclude the reliance of the mobile sensor nodes on non-adaptive GPS receivers in a network comprising of nodes in the order of hundreds to thousands or even more [14]. On the other hand, GPS-free localization algorithms for wireless sensor networks (WSN) [7, 9, 19] also exist in the literature which are estimation-based approaches localizing the nodes imprecisely.

Localization algorithms for mobile sensor nodes are also presented [10, 11] in WSN that obviously do not take into account the architectural heterogeneity of WSAN. Localizing sensors and actors independently, the individual location errors in sensors and actors are accumulated to large inaccuracy, which severely hampers the application performance. A localization protocol [2] for WSAN is also presented, which uses received signal strength indicator (RSSI) to calculate distance between beaconing actors and immobile sensors. Clearly, the RSSI-based approaches are highly sensitive to the environmental factors such as multipath reflections, non line-of-sight conditions, and other shadowing effects which lead to erroneous distance estimates [15]. Hence, there exist no algorithm that addresses the localization of mobile sensor nodes in WSAN, particularly when the anchors are mobile as well.

In this paper, timing-based localization algorithm of mobile sensor nodes (TMSL) is proposed that localizes the sensor nodes relative to the actor nodes. In TMSL, sensor nodes locally estimate the propagation delay of beacons, which are periodically transmitted by the actor nodes according to certain interval pattern. By following the transmission pattern of beacons, sensor nodes are able to determine the start time of beacons by themselves and use their own clocks to estimate the propagation delay. This approach does not require the timestamps of actors and, hence, avoiding the need of clock synchronization. The estimated propagation delay corresponds to the distance from the beaconing actor and measurements of three actors are used in *quadratic trilateration* method to compute the coordinates of the sensor node. Furthermore, the localization of mobile nodes generally requires continuous transmission of beacons that produces large overhead. TMSL also overcomes this problem by providing an adaptive solution that controls the beacon frequency according to the desired location accuracy to minimize the overhead. Hence, localization of mobile sensors relative to mobile actors, a novel technique to measure propagation delay, and adaptiveness to mobility features distinguish TMSL from the existing techniques. Simulation results prove that TMSL is adaptive to achieve the desired accuracy with controlled overhead.

The remainder of the paper is organized as follows. In Section 2, we review the existing localization approaches. We highlight the basic design principles, and present the architecture and algorithm of TMSL in Section 3. Error analysis of the proposed technique is made in Section 4 to determine its theoretical error bounds. Performance evaluation and simulation results are provided in Section 5. Finally, concluding remarks are discussed in Section 6.

2 Related work

The existing localization techniques in the sensor network literature can be categorized into three groups: range-based, range-free and mobility-aware techniques. In this section, we investigate the various proposals under these categories along with their drawbacks especially for the WSAN deployment scenarios where the sensor and actors both can be mobile.

2.1 Range-based localization

In range-based localization solutions, the absolute distance (range) between a reference node and a localizing node can be estimated by using one of these approaches; received signal strength, the time-of-flight of communication signal and angle of arrival of the received signal. However, the accuracy of such estimation depends on the surrounding environment and usually relies on complex hardware [16].

2.1.1 Received signal strength indicator

This technique requires the knowledge of transmitter power, the path loss model and the power of the received signal. Based on the transmit power and the path loss model known a priori, the effective propagation loss can be calculated. Theoretical and empirical models [13, 14] are used to translate this loss into a distance estimate. This method has been used mainly for RF signals. The major drawback of this method is that multipath reflections, non line-of-sight conditions, and other shadowing effects might lead to erroneous distance estimates [15].

2.1.2 Timing-based methods (ToA, TDoA)

This approach records the time-of-arrival (ToA), i.e., propagation time of the beaconing signal [12] or time-difference-of-arrival (TDoA) [17] of two signals that might be RF-signal and another signal of low propagation speed from a set of reference points. The propagation time can be directly translated into distance, based on the known signal propagation speed. A timing-based position scheme (TPS) [17] is proposed that takes the TDOA of three RF beacons and applies trilateration to compute the intersection of three circles of beaconing stations. However, this approach is not applicable for mobile sensor nodes, since a node can move to different position when the second beacon arrives producing significant error in propagation time. Use of ultrasound signal has also been suggested [16] as the second reference beacon for densely deployed sensor networks because of smaller propagation speed of ultrasound signals compared to RF. Although this gives fairly accurate results, it requires additional hardware on sensor nodes to receive the ultrasound signals. At the same time, this approach is range dependent due to short range of ultrasound transmission.

2.1.3 Angle-of-arrival

Special antenna configurations, e.g., omni-directional, are used to estimate the angle of arrival at which signals are received from a beacon node and use simple geometric relationships to calculate node positions [15]. The main drawback of this technique for terrestrial systems is the possibility of error in estimating the directions caused by multi-path reflections.

2.2 Range-free localization

To overcome the limitations of the range-based localization schemes, many range-free solutions [3–7, 9, 18, 19] have been studied. These solutions estimate the location of sensor nodes either by exploiting the radio connectivity information among neighboring nodes, or through applying some probabilistic approach. Among existing range-free localization approaches, ring overlapping comparison of received signal strength indicator (ROCRSSI) [3] technique uses ring overlapping to estimate nodes location. The rings can be generated by comparison of the signal strength a sensor node receives from a specific anchor and the signal strength other anchors receive from the same anchor. Then, a node calculates the intersection area of these rings and takes the gravity of this area as its estimated location. The ring overlapping, compared to triangle-overlapping in area-based point-in-triangulation (APIT) [4] technique, generates smaller intersection area and results in more accurate location estimation. The accuracy of APIT depends on the number of anchors, which surrounds the expected position of node. Higher the number of anchors, smaller the confined area that would provide a good location estimate. However, the accuracy of ROCRSSI is based on the assumption that in a certain range of direction, with the increase of distance between a sender and a receiver, the signal strength at the receiver decreases monotonically. It is highly dependent on the surrounding environment and density of nodes that may lead to inaccurate distance estimates [15, 24].

In mobile-assisted localization [5], a roving human or robot wanders through an area, collecting distance information between the nodes and itself. The challenge is to design movement strategies that produce a globally rigid structure of known distances among the static nodes. The pairwise node distances resulting from this strategy can then be fed into an anchor-free localization (AFL), which computes an initial coordinate assignment to all the nodes, using the radio connectivity information alone. This approach is highly dependent on the movement of mobile and degree of network, and hence, produces inaccurate results in a field where the movement of mobile is restricted and the deployment of sensor nodes is non-uniform.

A statistical beacon-less approach [6] assumes that sensors are deployed in the form of groups at different known regions. A node can estimate its position by observing group membership of its neighbors. The limitation of this approach is the deployment of sensor nodes in their related groups, which is possible only when deployed manually and, therefore, is not applicable in random deployment scenario. In addition, the location of a sensor node is determined based on its group, which does not provide accurate location. Another range-free solution, DV-hop [19], was proposed that exploits the connectivity of the nodes and determine the distance from anchor nodes by multiplying the hop count with an average hop distance. A drawback of DV-hop is that it fails for highly irregular network topologies, where the variance in actual hop distances is very large [8].

2.3 Localization of mobile nodes

Mobility of sensor nodes has also been addressed in literature [10, 11]. In [11], localization of mobile sensor nodes is based on the idea of mobile robot localization called Monte Carlo localization. The algorithm depends critically on movement of nodes and if the nodes stop moving, the algorithm can no longer gather useful data from seeds. In the polygon-based [10] technique, every node maintains convex polygon that represents its current estimate of the set of its probable locations and exchanges this information with its neighbors. It computes the intersection of every polygon it receives with its own polygon. The result of these intersections is the new location polygon for the node. Clearly, this approach consumes tremendous amount of energy since all the nodes participate in exchanging polygon information that will be very frequent at higher nodes velocity. Another problem of this approach is to maintain certain density of sensor nodes as well as seed nodes, which is a challenging task in mobile environment. Thus, the accuracy cannot be achieved if the mobile seed nodes are concentrated at a single point.

2.4 Contribution of TMSL

Clearly, there exists no unified and adaptive localization technique for WSAN in the literature that addresses the issue of localization of mobile sensors relative to actors and adaptiveness, i.e., provides a trade-off between localization accuracy and energy efficiency. Most of these algorithms are proposed for WSN which are inapplicable to WSAN. WSN has been widely used for monitoring the environment and the accuracy of localization can be compromised to some extent. In contrary, the exact location of the sensor nodes in WSAN assist the actors in executing precise actions. Therefore, the location accuracy of sensor nodes is indispensable with respect to the actors. At the same time, localizing the sensors and actors independently produce large positioning errors causing imprecise actions; thereby, minimizing the effectiveness of actors in the field. Thus, actors can be used as anchor nodes to neutralize the impact of any inaccuracy in location of actors and avoid special anchor nodes. Moreover, the mobility of sensor nodes necessitates an adaptive unified localization algorithm for WSAN that can be fine tuned to eliminate the errors caused by the mobility.

TMSL addresses the above issues and contributes in three fronts.

1. TMSL exploits the actor nodes which are always present in WSAN, and therefore, does not require or rely on external anchor nodes or seed nodes, particularly on the density of seed nodes.
2. Sensor nodes in TMSL consume much lesser energy since they only receive periodic beacons from the actors and do not transmit any message for localization. Even, the beacon frequency of actors is controlled to minimize the overhead as will be explained in Section 3.3.
3. TMSL algorithm considers the mobility of sensor nodes as well as actors and provides an adaptive approach that restricts the localization error to the given permissible tolerance level in order to control the potential localization overhead due to the mobility of nodes.

3 Timing-based mobile sensor localization

In this section, we present Timing-based Mobile Sensor Localization (TMSL) algorithm for WSAN. The objective of TMSL is to localize the mobile sensor nodes relative to the actors with minimum energy consumption of sensor nodes and achieve the required accuracy by limiting the distance errors within tolerable limits. It exploits the actor nodes as mobile beaconing reference stations. The heuristics of TMSL is its adaptiveness to network dynamics and flexibility to customize for the desired accuracy to limit the overhead incurred at higher node velocity.

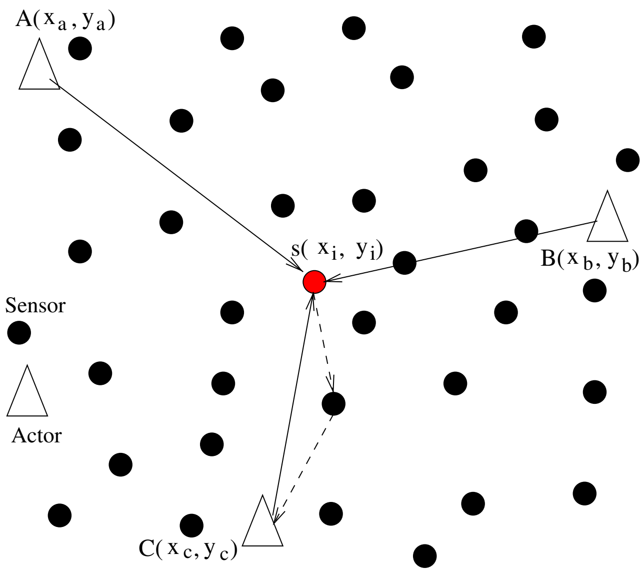


Fig. 1 Network model of TMSL, where actors *A*, *B* and *C* are used to localize the sensor node *s* at (x_i, y_i)

3.1 Network model

We assume that both the sensors and actors are mobile and randomly deployed in two-dimensional space. Actors and sensors are physically distinct nodes, where sensors monitor the sensing field and actors respond to the reported event. Actors have no limitation of energy and can transmit RF beacons with sufficient transmission power which are possibly received by all the sensor nodes in the entire field. Actors also function as *anchors* for localization of sensor nodes.[1]

Although more than three anchors can be present at any time, only three of them, i.e., A, B, C, with known coordinates (x_a, y_a), (x_b, y_b), and (x_c, y_c) as in Fig. 1, respectively, are selected by the sensor node, which are farther from the sensor position and non-collinear. We also assume that only the actors are synchronized with each other precisely by using synchronization algorithm [22, 23]. However, there is no requirement on the synchronization of sensor nodes.

3.2 TMSL overview

In TMSL, sensor nodes determine their distance from the actors by measuring the propagation time of the *reference beacons*. Usually the propagation time is measured by employing either round-trip-time mechanism [20] or establishing time base (synchronizing the nodes) [12]. However, the heuristic of TMSL is the local measurement of the beacon delay independent of the actors clocks.

To realize this approach, the transmission of beacons is scheduled according to the predefined pattern of intervals. This pattern is formed by dividing the certain *localization time period*[2] (τ) into discrete time units called intervals[3] whose duration is set according to the mobility of sensor nodes and presumed accuracy as described in Section 3.6. The value of τ reflects the possible topological changes either due to the mobility of nodes or incremental deployment of nodes. For network of stationary nodes having no future redeployment of sensor nodes, the localization time can be set to infinity to avoid potential overhead. On the other hand, τ is set to small value if the expected velocity of nodes is high. In such case, the mobile sensor nodes need to be re-localized frequently by transmitting the beacons. If the distance error is to be maintained smaller than some tolerable value, it means that a beacon must be received by the sensor node before the traveled distance approaches to the tolerable distance error. Hence, the selection of τ plays critical role in achieving the desired accuracy.

Four actors are assumed to transmit beacons in each interval. However, a sensor node, using trilateration method, selects three of them for which the localization error would be minimum as described in Section 3.5.2. The intervals are followed by the sensor nodes to locally compute the beacon propagation delay. In order to track the intervals, sensor nodes establish localization base time[4] (LBT) as they receive the first reference beacons from the actors at the beginning of τ.

Once LBT is determined, they can compute the start time of the beacons locally by using the interval information relayed by the actors. This information includes the interval number in which the beacon is transmitted and some lag time elapsed since the beacon was scheduled for its transmission. Hence, it makes the approach independent of clocks of the actors. For example, sensor *s* receives a beacon at time t_1. The beacon contains the interval number *i* and transmission lag time t_{lag}. It computes the total time $T(i)$ elapsed

[1]However, if the network is WSN instead of WSAN, then minimum of three sinks or sensors with the similar capabilities of these actor nodes, would function as anchors.

[2]Time period during which the nodes are localized and is divided into fixed number of intervals, which are numbered to compute propagation time. If nodes continuously keep moving then the maximum interval number limit can be reached. In such case, τ is reinitialized to restart interval numbering.

[3]An interval is a time period during which each actor transmits a beacon.

[4]Localization starting time or the time at which first interval starts.

until i^{th} interval and can infer the propagation time ($t_p = t_1 - (LBT + T(\text{i}) + t_{lag})$) using its local clock as explained in Section 3.5. Note that, only t_{lag} and the interval number are required to compute the propagation time.

The estimated propagation time of the beacon remains almost the same until the actor moves. Sensor nodes take the mean of all the measured propagation time values in calculating distance from the actor. However, if the actor moves then the propagation delay is changed. Therefore, the actor informs about its mobility to the sensor nodes notifying them to neglect the obsolete observations obtained before its mobility in order to compute the correct mean value. As sensor nodes compute propagation delay of beacons, they calculate distance and apply quadratic method of trilateration to find their position. If the distance estimates of the four beaconing actors are obtained, then they choose the triplet that has minimum root mean square deviation in distance estimates. Although, three beaconing actors are sufficient for the operation of TMSL, the availability of four beaconing actors is useful for selecting three farther actors that improves the localization accuracy.

3.3 Beacon transmission in TMSL

Here, beacon structure, its transmission rate and scheduling are described, which are the fundamental steps in TMSL.

3.3.1 Beacon packet format

The format of the beacon packets containing the required information to be used by sensors in TMSL operation is shown in Fig. 2. A beacon carries the following information which are mandatory for the sensors to calculate the propagation time.

- *Interval No.*: To achieve the varying beacon frequency for different applications or scenarios, either τ can be fixed and divided into different number of intervals or the number of intervals are fixed but τ is varied. TMSL fixes the maximum number of intervals since the actors insert the interval numbers in beacons and variable interval limits for different scenarios would require a varying beacon size, which is not possible. Hence, the interval space of 15 bits is fixed to keep track of the current interval. If lesser size is reserved then it results in smaller interval limit requiring to initialize the intervals sooner. On the other hand, reserving more bits in packet results in extra overhead.
- *M flag*: In order to isolate the errors in timing measurements caused by the external environment, TMSL computes the mean of the estimated propagation time of beacons over multiple intervals during which actors are stationary as described in Section 3.5.1. As an actor moves, it propagates the movement information by setting the flag M in the next scheduled beacon. Sensor nodes ignore the previously measured values in computing mean of the propagation time if the flag M in the beacon is set.
- *Lag time*: Although the beacon is scheduled to be broadcast at predetermined time, transmission of the beacon may be delayed due to some factors, e.g., medium access time and processing time. These are accumulated as lag time and actors report this value in the beacon for precisely measuring the time-of-flight of the beacon.
- *Excessive time*: Each interval is divided into four slots for beacon transmission of four actors and actors transmit beacons in their respective slots in some order. To establish LBT, each actor reports the excessive slot time of its preceding actor, i.e., time from beacon reception to slot period expiry, referred to as excessive time.
- *Actor Position*: X and Y coordinates of the actor are stamped in the beacon to form the overlapping rings to be used in position estimation of unlocalized nodes.

16 bits	1 bit	32 bits	32 bits	32 bits	32 bits
Interval no.	M	X–coordinate	Y–coordinate	Lag time	Excessive time

Fig. 2 Packet format of *reference beacon*

3.3.2 Timing of intervals

The number and duration of intervals are determined by the localization time period τ, which in turn is based on the expected mobility of nodes or how frequent the network topology may change. The interval information is required by the sensor node to determine the start time of beacons. It is important to note that this does not include any clock information of the actor nodes. Hence, the propagation time is measured by taking the time difference of the start time measured locally by the sensor node and beacon received time.

By reserving 15 bits for interval numbering in beacon, τ can be divided into 2^{15} number of intervals of duration $\tau/2^{15}$. Therefore, beacon transmission frequency, i.e., f_b, is

$$f_b = 2^{15}/\tau \tag{1}$$

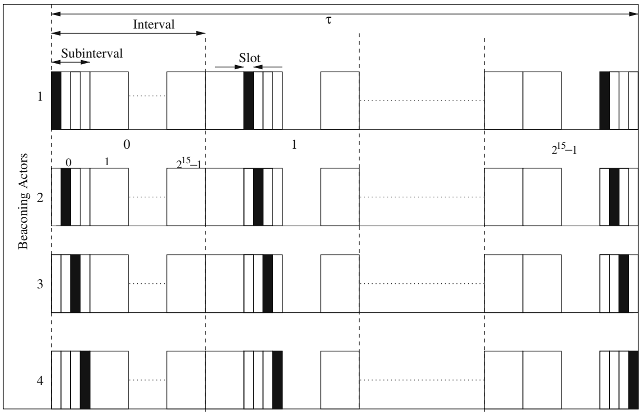


Fig. 3 TMSL timing information for *reference beacon* transmission by four actors

Furthermore, each interval is divided into 2^{15} subintervals to increase the precision of timing information as illustrated in Fig. 3. Therefore, the time period of beacon transmission is the sum of time period of an interval and a subinterval, which is $\tau/(2^{15}+2^{30})$. Let $T(i)$ be defined as timing function, which determines the actual start time of i^{th} beacon transmission. Then, $T(i)$ can be expressed by

$$T(i) = (i-1)\left[\frac{\tau}{2^{15}} + \frac{\tau}{2^{30}}\right] \quad (2)$$

From (2), the 1^{st} beacon is transmitted in 0^{th} interval at $T=0$, the 2^{nd} beacon in the first interval at $T=(\tau/2^{15}+\tau/2^{30})$ and the 3^{rd} beacon at $T=2(\tau/2^{15}+\tau/2^{30})$ and so on.

Although the beacons are scheduled to be transmitted by each actor at the time calculated by (2), it is not possible that all the actors capture the shared channel at the same time for beacon transmission. If the synchronized actors attempt to access the medium simultaneously, their transmission will collide that would result in higher contention delay. Such unexpected higher delay may have more severe impacts, if it exceeds the interval duration. Consequently, if the mobile nodes could not receive beacon within the interval then, they will either remain unlocalized or keep incorrect position until beacon of corresponding interval is received.

MAC contention delay is an important factor when computing the propagation delay that causes inaccuracy in distance calculations. If the sink node is present in the network, then it broadcasts the schedule to actor nodes. Otherwise, they decide by their IDs in the increasing order. Therefore, the subinterval is shared among four actors by dividing the subinterval into four time slots as shown in Fig. 3. The length δ of the slot can be computed as

$$\delta = 2^{30}/4$$

Each actor waits for its turn in the subinterval and utilize its allocated slot for transmission. Hence, the transmission time vector for four actors in i^{th} interval is $[T(i), T(i)+\delta, T(i)+2\delta, T(i)+3\delta)]$.

3.4 Localization base time

Following the beacon transmission schedule, TMSL establishes localization base time (LBT) for the sensor nodes. LBT is based on the beacons received from the first two actors. Let s be the sensor node determining its localization base time. We assume that A is the first actor to transmit beacon in the interval and B is the second actor that transmits beacon following A in the same interval. Now, let s hear the beacons from actors A and B at time t_1^s and t_2^s, respectively, as illustrated in Fig. 4.

Due to the synchronization of actors, actor B infers the departure time t_1 of the beacon transmitted by A. Let B hear the beacon from A at time t_1^{b} such that $(t_1^{\mathrm{b}} - t_1 < \delta)$, where δ is the slot length, i.e., proportion of time reserved for an actor to transmit its beacon. Sensor node s receives beacon earlier than actor B since actors are initially placed at the corner of the field and are farther apart than sensor nodes. Let the time that B receives beacon before the slot length of A expires, i.e., excessive time, denoted by $t_{\mathrm{exc}}^{\mathrm{a}}$. Therefore, $t_1^{\mathrm{b}} - t_1 + t_{\mathrm{exc}}^{\mathrm{a}} = \delta$. Knowing δ, t_1, and t_1^{b}, actor B computes $t_{\mathrm{exc}}^{\mathrm{a}}$ and inserts it in its beacon packet in addition to the lag time of beacon transmitted by B. In the following slot, actor B transmits its beacon which is received by sensor node s at time t_2^s and s computes the round trip time between s and B as $\left(t_2^s - t_1^s - t_{\mathrm{lag}}^{\mathrm{b}} - t_{\mathrm{exc}}^{\mathrm{a}}\right)$ based on beacon of A received by s and B, and beacon of B received by s as illustrated in Fig. 4. Hence, the estimated

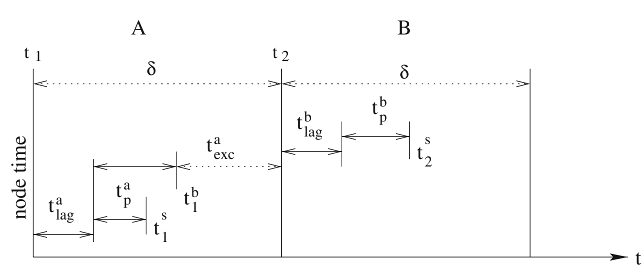


Fig. 4 Determining localization base time of sensor

propagation time t_p^b of the beacon received from B is obtained

$$t_p^b = \left(t_2^s - t_1^s - t_{lag}^b - t_{exc}^a\right)/2 \quad (3)$$

It can be noted that t_2^s and t_1^s are the time locally observed at node s, while, t_{lag}^b and t_{exc}^a are the values reported by actor B in its beacon. Hence, s can deduce that the slot time of B starts at $\left(t_2^s - \left(t_p^b + t_{lag}^b\right)\right)$ of its local clock. Since each actor takes one time slot for beacon, the starting time of interval or localization base time (LBT_s) of sensor node s measured with k^{th} actor is obtained

$$LBT_s = t_2^s - \left(t_p^b + t_{lag}^b + k * \delta\right) \quad (4)$$

where $k \leq 4$ and is the number of the actor transmitting the second beacon. Here, if B is the immediate follower of A then k will be 2. In this way, all the sensor nodes determine their own LBT in first interval.

3.5 Node localization

The establishment of LBT provides basis to compute propagation time and is fundamental for node localization. The node localization procedure is composed of two phases: *distance measurement* and *node positioning*. In the first phase, sensor nodes measure distance from all actors that are used to estimate the locations of nodes during the second phase.

3.5.1 Distance measurement

Given the propagation time of a beacon from sender to receiver, the distance between the two nodes can be measured by using propagation speed of the RF signal, i.e., the speed of light. In TMSL, sensor nodes follow the same approach to measure the distance D from actors, i.e.,

$$D = c \times \bar{t_p} \quad (5)$$

where c is the speed of light that is approximately *299792458 m/s*. $\bar{t_p}$ is the mean of t_p values obtained over the intervals when sensor and actors are stationary. To determine t_p, actor nodes include the current interval number ($\Im$) in their beacons. As the sensor node receives beacon, it computes the start time $t_s = \tau_o^s + T(\Im)$ of the beacon. Likewise, every node receiving the beacon is able to estimate t_s. The propagation delay t_p of the beacon is obtained as

$$t_p = t_r - t_s$$

where t_r is the received time of the beacon.

Usually, the transmission of beacon does not start at its scheduled time. It is due to the fact that the wireless channel might be contended among large number of sensor and actor nodes that prevents it to capture the channel immediately. This causes some lag in the actual beacon transmission. Actors report this lagging time t_{lag} in beacons to accurately measure t_p. It is highly likely that they measure incorrect t_p if t_{lag} is not reported. The beacon carries transmission delay in its *Lag Time* field and is calculated as

$$t_{lag} = \delta \times slot_{no} + t_{mac}$$

where $slot_{no}$ is the slot number allocated to each actor in subintervals and $\delta \times slot_{no}$ is considered as scheduling delay. In addition, the medium may still not be available due to the transmission of surrounding sensor nodes or captured by some actor node, which results in medium contention delay t_{mac}. The value of t_{lag} reported in beacon is subtracted from the estimated value of t_p to obtain the accurate time-of-flight as:

$$t_p = t_r - t_s - t_{lag} \quad (6)$$

In order to obtain the mean of the propagation time, TMSL keeps recording the values of t_p until either the actor moves or node itself. The movement of actor is reported in beacon by setting the flag M, while the nodes are aware of their own mobility. Let collecting the samples of t_p starts at time t and suppose that at time $\acute{t}$, either actor moves or node itself. Assume that a node receives k beacons from a particular actor during the time period $\acute{t} - t$ and measures the instantaneous beacon propagation delays $t_{p1}, t_{p2}, ...t_{pk}$ for each beacon. The sample mean over the k intervals can be found as

$$\bar{t_p} = \frac{1}{k}\sum_{i=1}^{k} t_{pi} \quad (7)$$

This implies that the accuracy of the sample mean as an estimator of the t_p mean increases with the sample size k. Hence, an unbiased estimator of t_p variance becomes

$$\hat{t_p} = \frac{1}{k-1}\sum_{i=1}^{k}(t_{pi} - \bar{t_p})^2 \quad (8)$$

The deviation $\hat{t_p}$ is considered in the first few values of t_p to minimize the possible errors in measurements after either the actor moves or node moves itself.

3.5.2 Spatial location of nodes

In order to compute the location of sensor nodes by using positioning system, there are three commonly used techniques: triangulation, trilateration and multilateration. Triangulation is based on Angle-of-Arrival (AoA) that requires precise antenna configuration. Multilateration can improve accuracy but involves higher computation overheads. TMSL uses the trilateration approach in which three actors are selected to work as beaconing nodes. Trilateration is an approach to determine the position of an unlocalized nodes based on range measurements from three beacons located at known positions. In addition, there are three methods for the trilateration: quadratic equation method which is used to find an intersection of three spheres, Cayley–Menger determinant method [25] which uses the geometric relationship between three beacon positions and a listener position, and nonlinear least squares method which minimizes uncertainty [26]. In [21], hybrid trilateration approach is applied in order to minimize the errors in different configuration of three circles. However, due to the complexity of hybrid approach, which is not suitable for energy constrained devices, TMSL adopts quadratic equation method which yields sufficiently high accuracy with minimum computational overhead.

TMSL algorithm assumes distance estimate of four actors, while trilateration method requires three known points. It gives a choice to select three suitable anchors for which the localization error is minimum using quadratic method. Clearly, it is observed in Fig. 5 that the error is minimum when the unknown point is close to the center of three points. The error increases as the unknown point gets closer to any of the known point or moves far from the center. In TMSL, sensor nodes compute the root mean square deviation in distance

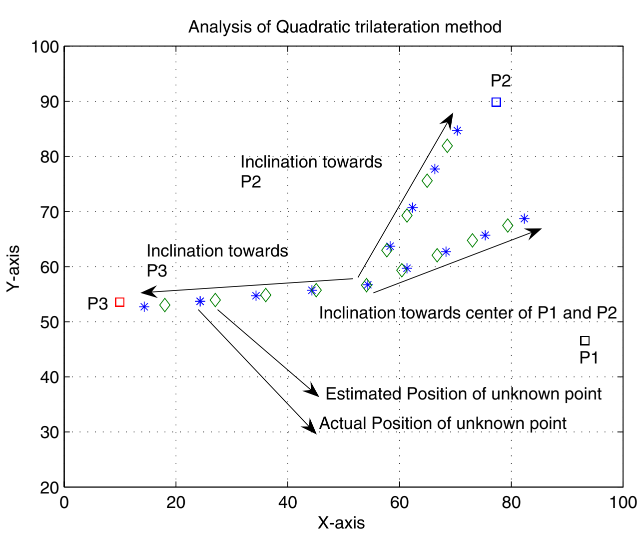


Fig. 5 Analysis of quadratic method of trilateration, where P1, P2 and P3 are known points. The unknown point $*$ is taken at different positions to report the error in quadratic method at 5% error in distance estimates of unknown point

(σ_r) for all the possible combination of three actors and select the triplet for which the deviation is minimum.

Let A, B and C be the three actors which have minimum σ_r and (x_a, y_a), (x_b, y_b) and (x_c, y_c) be their positions, respectively. Sensor node i estimates distance r_a, r_b and r_c from these actors, respectively, by using (5). Let (x_i, y_i) be the position of the node i, which is unknown. By using the Euclidean distance formula, following second order distance equations can be obtained

$$(x_i - x_a)^2 + (y_i - y_a)^2 = r_a^2 \tag{9}$$

$$(x_i - x_b)^2 + (y_i - y_b)^2 = r_b^2 \tag{10}$$

$$(x_i - x_c)^2 + (y_i - y_c)^2 = r_c^2 \tag{11}$$

The problem of intersection of three circles can be easily simplified to that of obtaining a line and a circle. The intersection point of line and circle gives the approximate position of unknown point.

Equations (9), (10) and (11) can be simplified into two linear equations. By subtracting (14) and (15) from (13), we obtain

$$x_i(x_b - x_a) + y_i(y_b - y_a) = \left(r_a^2 - r_b^2 + v_{ab}\right)/2 \tag{12}$$

$$x_i(x_c - x_a) + y_i(y_c - y_a) = \left(r_a^2 - r_c^2 + v_{ac}\right)/2 \tag{13}$$

where $v_{ab} = x_b^2 + y_b^2 - x_a^2 - y_a^2$ and $v_{ac} = x_c^2 + y_c^2 - x_a^2 - y_a^2$. The values of x_i and y_i can be obtained by solving (16) and (17) as follows:

$$x_i = \frac{k_1 - (y_b - y_a)y}{x_b - x_a} \tag{14}$$

$$y_i = \frac{k_2(x_b - x_a) - k_1(x_c - x_a)}{(x_b - x_a)(y_c - y_a) - k_2(x_c - x_a)} \tag{15}$$

where $k1 = \left(r_a^2 - r_b^2 + v_{ab}\right)/2$ and $k2 = \left(r_a^2 - r_c^2 + v_{ac}\right)/2$.

Hence, each node determines its coordinates (x, y) at the end of each interval. The values of x and y may have error whose magnitude depends on the location of sensor node in the field and, therefore, is different for nodes located at different positions as analyzed in Section 4. Moreover, these coordinates also become inaccurate if the velocity of mobile nodes is higher than the beacon frequency (f_b). In the next section, we establish a relationship between the predicted velocity (v) of sensor nodes and f_b to control the inaccuracy due to the mobility of nodes.

3.6 Accuracy in TMSL and parameter selection

TMSL aims to provide accuracy according to the desired level for both the stationary as well as mobile nodes. Accuracy of the algorithm depends on the selection of localization period τ. This value is used to control the beacon frequency that should be adaptive to the velocity of nodes and network dynamics. If the nodes are mobile and move with high velocity then τ should be set to small value so that the beacon transmission frequency f_b is increased. On the other hand, the value of τ is selected large for stationary nodes since their position remain unchanged throughout the life of network once localized. However, the beacon frequency is also adjusted according to network dynamics due to the redeployment of nodes apart from the mobility. Therefore, this factor should be considered as well in deciding the value of τ. For example, if we take $\tau = 10\,h$ then the value of f_b is approximately 0.91 beacons per second using (1) and nodes can renew their spatial location after 1.098 s.

In case of incremental deployment of stationary nodes, if nodes are re-deployed after every 10 s then the *reference beacon* must be received by the nodes at least once in 10 s, so that the re-deployed nodes can be localized. In this case, the required frequency might be 0.1 beacons/s, ignoring the mobility of nodes, and therefore, τ can be set to approximately 9 h.

Similarly, improper selection of τ causes inaccuracy in location determination. For instance, when the velocity of a mobile node is 10 m/s. It changes its location by 10 m in a second. If the frequency of *reference beacon* is one beacon per second, then the maximum location inaccuracy is 10 m. However, by increasing f_b to ten beacons per second, we reduce the inaccuracy to 1 m. The high value of f_b results in high cost in terms of energy and bandwidth. Therefore, the location accuracy increases with large value of f_b, but decreases with small value of f_b at higher speed. Hence, there is a trade-off between accuracy and communication cost. High level of accuracy requires more resources but low accuracy is cost-efficient. The accuracy can be set according to the application requirements.

We can establish a relation between the accuracy and τ that helps in evaluating the performance of TMSL algorithm. Let η be the required accuracy level with the predicted velocity v of mobile sensor nodes, we can derive a relationship between f_b, η and v as:

$$f_b = \eta \times v \tag{16}$$

$$\tau = 2^{15}/f_b \tag{17}$$

The value of v is the predicted velocity of nodes rather than the actual velocity and may be different than the actual speed of sensor nodes. Hence, the TMSL protocol parameters can be set to achieve the desired accuracy and a proper choice of the beacon transmission frequency is essential in order to minimize the communication overhead as well as location errors. Therefore, using (16) and (17), TMSL may update the value of τ relative to the value of f_b rather than setting some fix value.

4 Error analysis

The initial configuration of actors is shown in Fig. 6 in which actors are placed at the corners of the field, while sensor nodes inside the field. Let t_p^{as} and t_p^{bs} be the propagation delays measured by sensor node s_1 for beacons transmitted by actors A and B, respectively, and t_p^{ab} be the propagation delay of beacon of actor A measured by actor B. Since propagation delay corresponds to the distance between two nodes, this implies that $t_p^{ab} \leq \left(t_p^{as} + t_p^{bs}\right)$, i.e., the propagation delay from actor A to actor B is less than or equal to the sum of the propagation delays from actor A to sensor s_1 and actor B to sensor s_1. It is due to the fact that the distance d_{ab} between the actors A and B is smaller than the sum of the individual distance of s_1 from the actors A and B as shown in Fig. 7a. In our measurements, the difference of this propagation delay results in localization error (χ). The value of χ is approximately zero when sensor s_1 lies on the transmission path of actors A and B. Figure 7b demonstrates this fact, which is the best case in determining LBT. The distance of s_1 from the

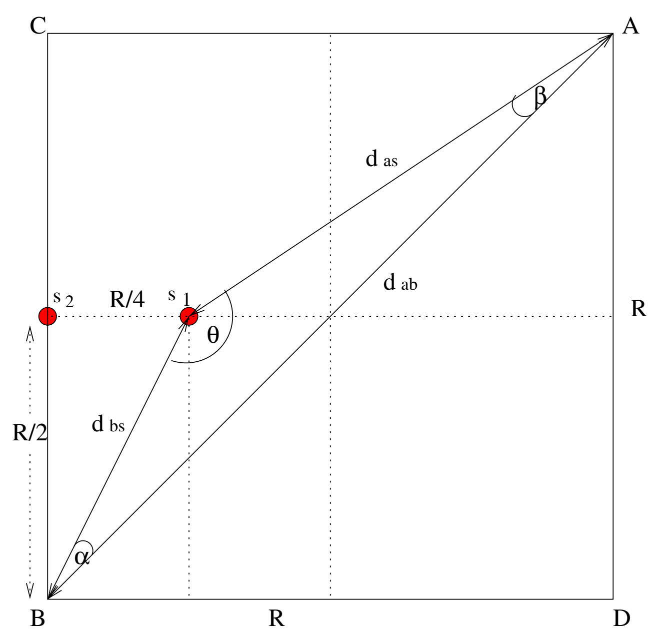
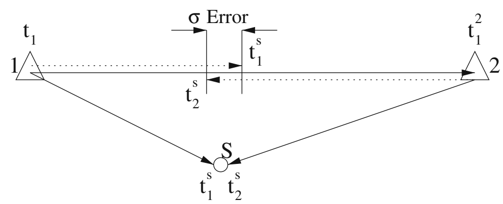
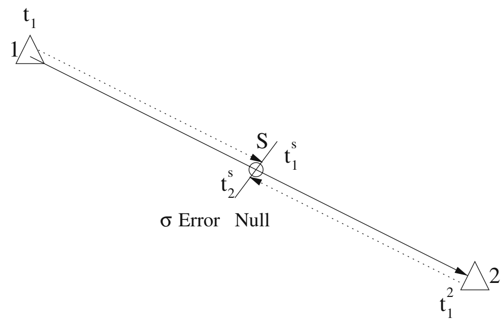


Fig. 6 Determining the least value of θ to analyze error in LBT



(a) General case of finding LBT.



(b) Best case of finding LBT when the sensor s lies on the path of actors 1 and 2.

Fig. 7 Computing LBT in different scenarios

transmission path of two actors can be translated into χ, where

$$\chi = (d_{as} + d_{bs} - d_{ab})/2 \tag{18}$$

From (18), it is clear that χ can be minimized by increasing d_{ab}, which is achieved by selecting the actors A and B such that they are the farthest to each other. This is possible when they are at diagonal vertices of the field, i.e., either at $(0, 0)$ and (R, R) or $(0, R)$ and $(R, 0)$ as in Fig. 6, where R is the length of the square field. Hence, the four actors are placed at the four corners of the field and sensors select a pair of diagonally positioned actors or any pair of actors for which χ is minimum. With this placement, it can be deduced that $d_{ab} = R\sqrt{2}$ and two pairs of diagonally positioned actors, AB and CD are present at locations $(0, 0)$, (R, R), $(0, R)$ and $(R, 0)$, respectively. Among the four quadrants of the filed, where the length of each is $R/2$, χ is most likely smaller for the sensors in quadrants I, III and II, IV with the pairs AB and CD, respectively.

Another way to reduce the value of χ is to decrease either d_{as} or d_{bs}. This can be achieved by increasing the value of θ. Applying law of sine in Fig. 6, we have $d_{as} = d_{ab}\frac{\sin(\alpha)}{\sin(\theta)}$ and $d_{bs} = d_{ab}\frac{\sin(\beta)}{\sin(\theta)}$. The best choice of a pair of actors is to select the actors such that θ is higher no matter they are diagonal or along the same side. Therefore, in Fig. 6, the choice of sensor s is the pair BC and choice of sensor s_2 is pair AB since larger value of θ leads to lower χ. On the other hand, smaller value of θ results in higher χ. For $\theta = 180$, the value of χ becomes 0 as is the case for s_2 with pair BC. Figure 8 shows the actual positions of sensor nodes as well as estimated using TMSL. It is observed that the nodes which lie close to the straight path between any pair of actors, where $\theta \approx 0$, the estimated location overlaps the actual location of sensor nodes otherwise the estimated locations are far apart.

We can now analyze the error in the above configuration as shown in Fig. 6. Here, $\alpha = 45^o - \cos^{-1}\left(\frac{R/2}{d_{bs}}\right) \approx 18.46$, where $d_{bs} = R\sqrt{3}/2\sqrt{2}$ according to the described setup. Hence, d_{as} can be computed as

$$d_{as} = d_{bs}^2 + d_{ab}^2 - 2d_{bs}.d_{ab}.\cos(\alpha) \tag{19}$$

Substituting the values in (19), we obtain

$$d_{as} = \left[\left(\frac{R\sqrt{3}}{2\sqrt{2}}\right)^2 + (R\sqrt{2})^2 - R^2\sqrt{3}\cos(18.46)\right]^{1/2} \tag{20}$$

$$= 0.85 \times R \tag{21}$$

Therefore, the maximum error that can be produced in this approach can be found by putting values in (18).

$$\chi = \left(\frac{R\sqrt{3}}{2\sqrt{2}}\right) + 0.85 \times R - R\sqrt{2})/2 \tag{22}$$

$$\chi = R \times 0.05 \tag{23}$$

Thus, the maximum localization error in TMSL is 5% of the field length R. It is close to 0 in the best case as illustrated in Fig. 7b, where the sensor node lies on the transmission path of the two actors. Hence, the error bound in TMSL is $[0, 0.05R]$.

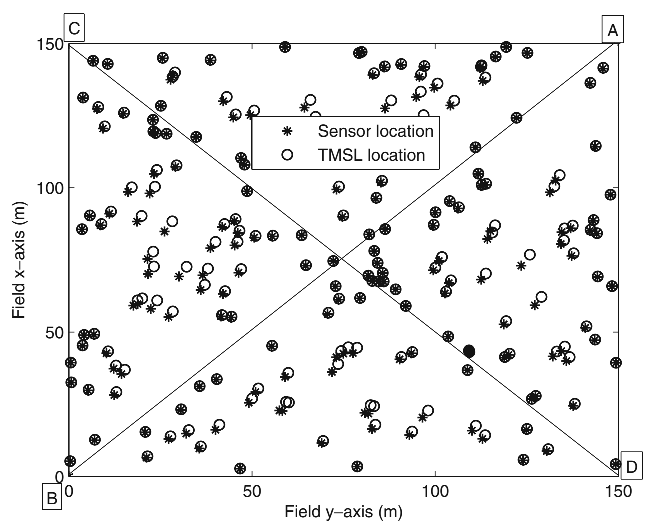


Fig. 8 Nodes actual location and estimated by TMSL

5 Performance evaluation

The performance of $TMSL$ is evaluated by using the network simulator *ns-2* [27]. Example scenario of wireless sensor and actor consists of 200 sensors deployed randomly in a field of 150 × 150 and four actors initially deployed at the corners of the field such that they lie in four different quadrants of the field with respect to the sensor nodes. The transmission power of actors is adjusted so that all nodes can hear *reference beacons*. As discussed in Section 3.6, selection of τ plays an important role in achieving accuracy. Therefore, TMSL performance is evaluated with fixed values of τ as well as adjusted according to application-specific localization accuracy η and predicted nodes velocity v, which is different from the actual speed of sensor nodes. Speed of sensor nodes reported in the experiments is the maximum value and nodes move at random speed between zero and this maximum limit. We run experiments for different scenarios by varying speed of sensor nodes, nodes density, speed of actors and bit error rate.

5.1 Mobility of sensor nodes

In this section, we analyze the impact of mobility of sensor nodes on location accuracy and how different TMSL parameters can be tuned to minimize the mobility effect. Figure 9 illustrates the mean localization error of sensor nodes for some fixed values of τ. It can be observed that the localization error increases as the speed of sensor nodes increases. However, these errors can be controlled by decreasing the value of τ. By decreasing τ from 40 to 30 h, i.e., by 25%, the location error is reduced approximately to 15%. On the other hand, the accuracy can not be maintained with the increased speed of sensor nodes. This is improved by setting η to some appropriate value. Figure 9 shows that the error at $\tau = 40$ h is reduced to 32% when η is set to 0.8.

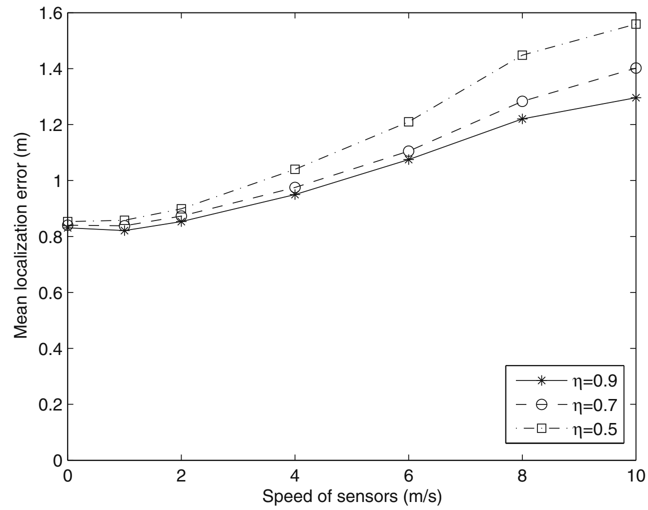


Fig. 10 Location error for different values of sensor node velocity at different values of η

Due to adaptiveness of TMSL, τ can be adjusted on the basis of two parameters: by predicting the maximum velocity v of sensor nodes to minimize the localization error and setting η to achieve application-specific accuracy with controlled overhead. The beacon frequency (f_b) is determined by using and, consequently, τ is obtained. It is shown in Fig. 10 that localization error decreases by increasing the value of η

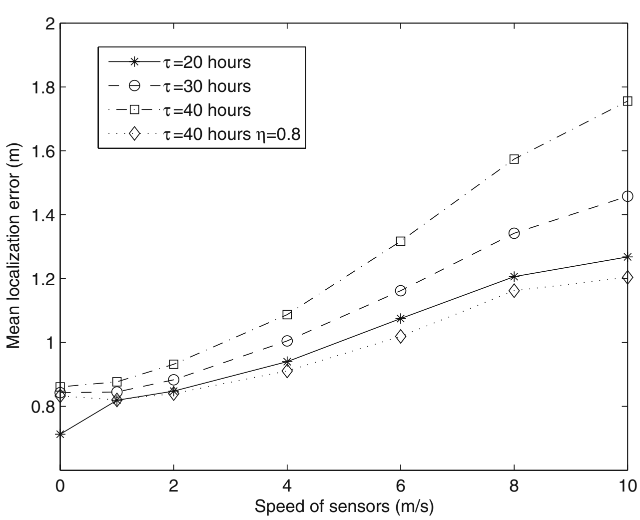


Fig. 9 Location error for different values of sensor node velocity at different values of τ

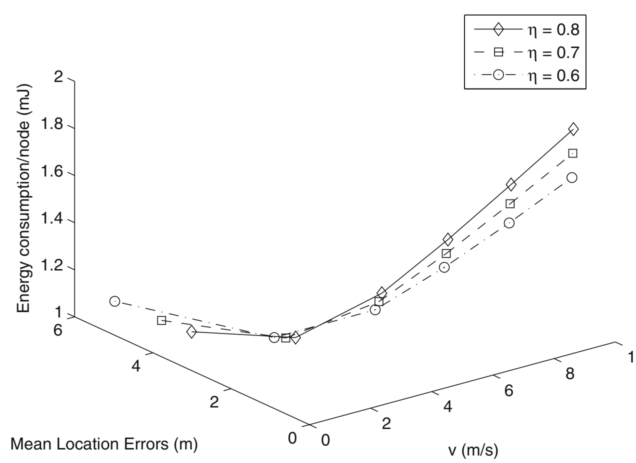


Fig. 11 Location error and communication cost with different values of f_b set according to η and different presumed values of v. Sensor nodes are moving randomly at maximum speed of 50 m/s

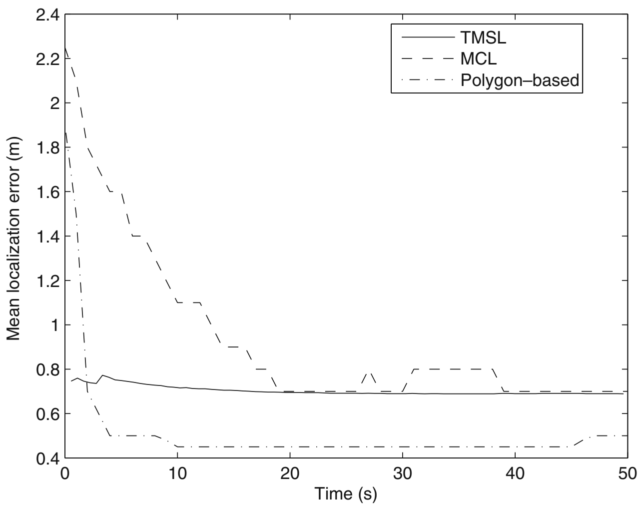


Fig. 12 Error comparison between TMSL and the existing techniques MCL [11] and polygon-based approach [10]. In this scenario, the sensor nodes move at the maximum speed of 20% of their transmission range

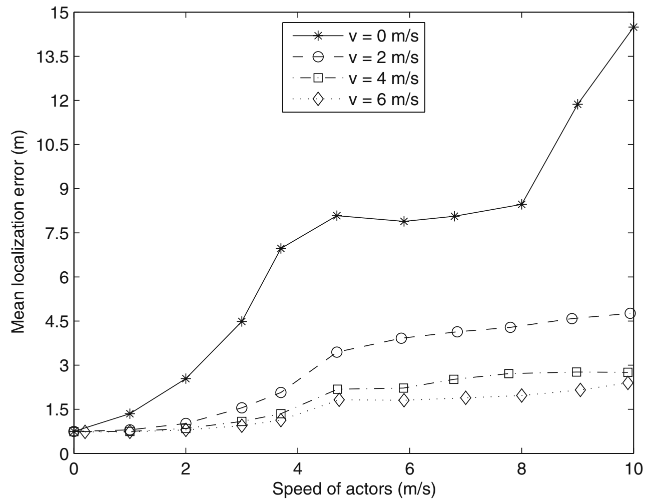


Fig. 14 Impact of mobility of actor nodes on localization accuracy at different values of v for stationary sensor nodes

but increases as the speed of sensor nodes increase. However, error due to the mobility of nodes can be minimized when v approaches to the actual speed of sensor nodes. Since f_b increases with v that causes higher communication cost, the value of η is set according to the required accuracy that limits the overhead. Figure 11 represents the relation of three factors, i.e., energy consumption, v and mean location error at different values of η. Note that the energy consumption reported is the total cost of all actors and sensors for a localization duration of 100 s. In our model, actor node consumes 0.23 mJ in transmitting a *reference beacon* over the entire field and nodes consume 0.56 μJ in receiving the beacon.

There exists no localization protocol which considers the mobility of sensor nodes as well as actors in WSAN.Therefore, a comparison is made between TMSL and some of the existing techniques Monte Carlo Localization (MCL) [11] and Polygon-based localization [10] proposed for mobile sensor nodes in WSN. The accuracy of these techniques depends on the density of seed nodes and suggested that at least one seed node should be present for a group of ten neighboring sensor nodes. However, there is no such restriction in TMSL that needs four actors regardless of the number or density of nodes. Another important

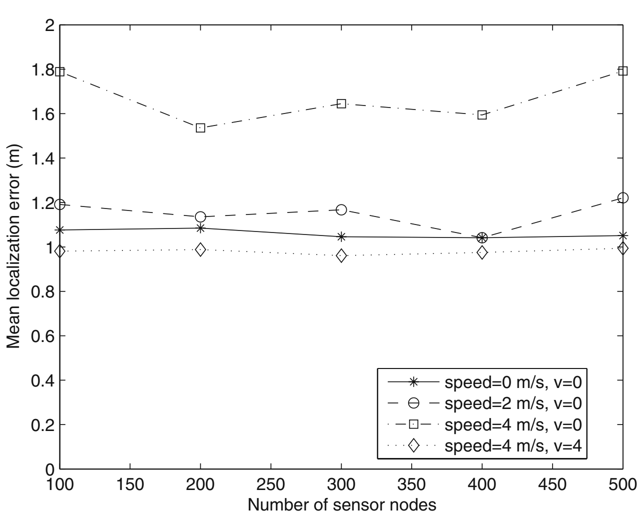


Fig. 13 Location errors at different sensor nodes speed for varying nodes density

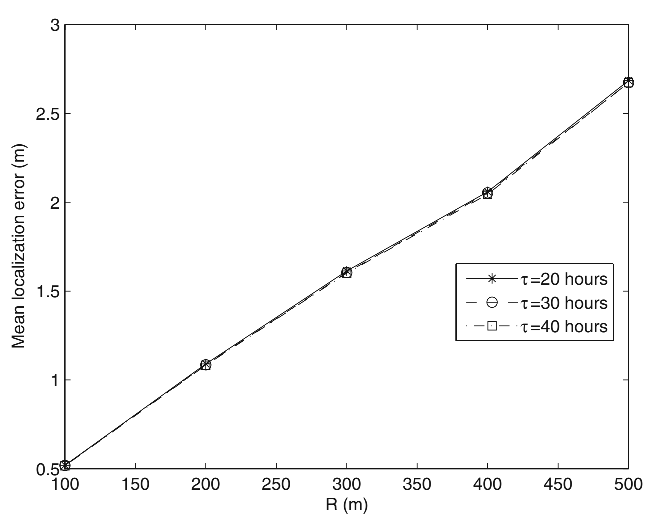


Fig. 15 Location error at different values of τ for varying the deployment field dimension with fixed 200 nodes

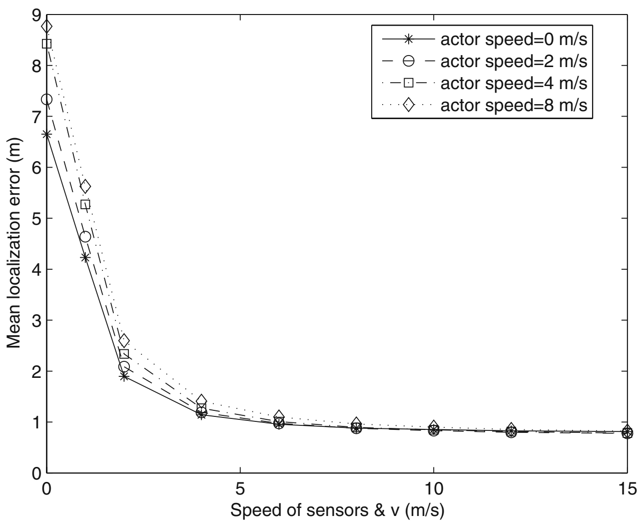


Fig. 16 Localization error for different speed of actors and sensors (v is taken equal to the maximum speed of sensor nodes

difference is the convergence time that these algorithms require to achieve suitable accuracy as shown in Fig. 12. In TMSL, nodes are localized as soon as they receive first beacon from the actors. On contrary, sensor nodes in [10] require some rounds to exchange the updated polygon information with each other to converge to accurate results. TMSL has slightly higher error of about 0.75 m than polygon-based technique that has 0.45 m. However, TMSL significantly outperforms polygon-based approach in terms of communication overhead since sensor nodes do not transmit any message for localization.

5.2 Network density

The scalability of protocol is measured for varying the number of nodes in a given deployment field. Figure 13 shows that the error remains approximately the same with slight fluctuation. At the same time, it is observed that the error is not a function of number of nodes but speed of nodes. Although the error is higher at higher speed but remains same for varying the nodes density. If we expect higher velocity of nodes, then taking v corresponding to that speed minimize the error which is applicable even to higher density. For $v = 4$ in Fig. 13, TMSL achieves the accuracy of 80% compared to the case where $v = 0$ when nodes are mobile. Hence, TMSL achieves high accuracy by adapting the protocol parameters for wide range of network density.

5.3 Mobility of actors

TMSL assumes that the sensors (unlocalized nodes) as well as the actors (anchor nodes) can be mobile and localization errors are equally controlled. Localization error for varying the speed of actor nodes at different settings of v is shown in Fig. 14. Similar to the mobility of sensor nodes, localization error due to the mobility of actor nodes is also minimized by predicting the expected velocity of actor nodes. As the predicted value is close to the actual speed of nodes, i.e., for $v = 4$, error is reduced approximately four times. This difference becomes more significant if the speed of actor nodes is higher. However, nonzero error in TMSL always exists regardless of the values of algorithm parameters τ, η

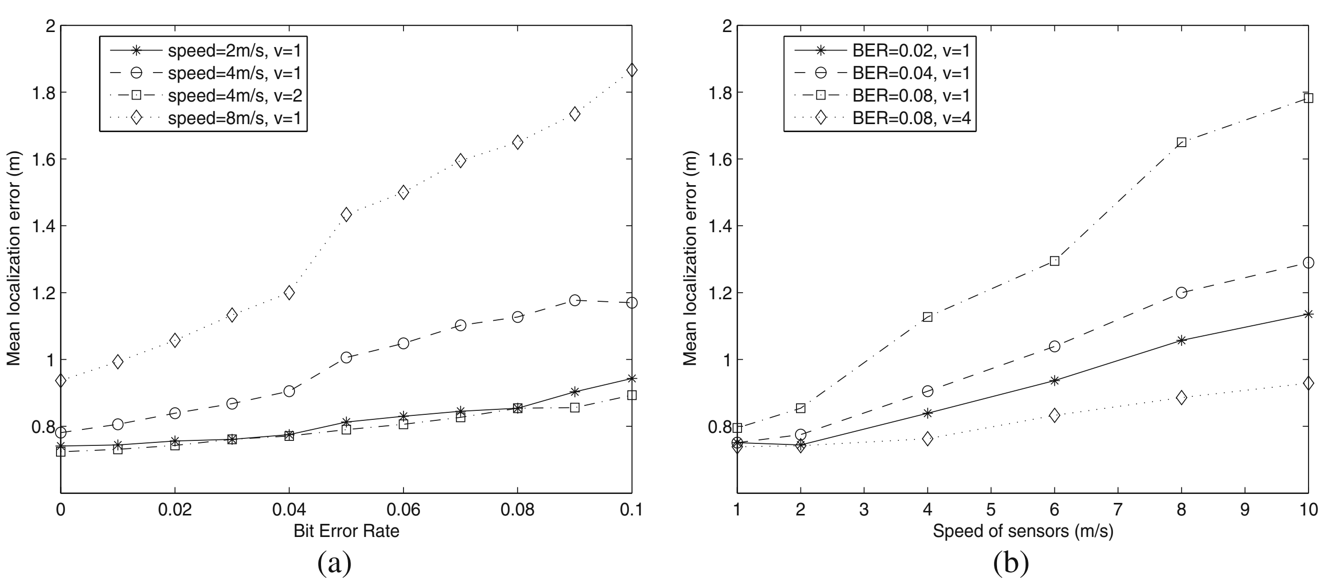


Fig. 17 Location error for **a** different speed of sensors and v with varying BER, **b** varying BER in addition to sensor mobility

and v. As the error analysis in Section 4 also shows, it is proportional to the dimension of the field as also observed in Fig. 15.

The simultaneous mobility of sensor nodes and actors is also considered in the experiments. It is observed that v is equally effective to the mobility of both kinds of nodes, which is depicted in Fig. 16. For a given value of v, the error in this case is almost the same as in the scenario of stationary sensors and mobile actors. For instance, the error in Fig. 14 for $v = 4$ and actor speed of 4 m/s is 1.35 m. The error for the same parameters in Fig. 16 is reported 1.27 m, which is an insignificant difference.

5.4 Bit error rate

Operation of TMSL depends on the beacons received from the four actors. In case of hostile environment, beacons may be lost or damaged due to higher bit error rate that may result in lack of updated information at sensor nodes and eventually cause higher localization error. Therefore, it is important to evaluate the algorithm for varying bit error rate (BER). Figure 17a shows the relationship of BER and location error, which is proved to be directly proportional. The results are plotted at different mobility of sensor nodes that adds up its effect to produce higher error. Basically, higher BER results in loss of beacon packets in some intervals that obviously refrains the sensor nodes to compute their locations during those intervals and therefore, they keep incorrect location. This effect is similar to the higher speed of nodes in which sensor nodes change their location and late arrival of beacon at low f_b causes them to keep old location information.

TMSL overcomes this loss of beacon packets by increasing its beacon frequency, which depends on v and η. By taking v proportional to the loss of packets apart from speed of sensor nodes, f_b is increased accordingly and eventually sensor nodes receive beacons more frequently. Consequently, the TMSL algorithm is invoked earlier allowing the nodes to compute their location frequently. By increasing the predicted value of v from 1 to 2 at sensors speed of 4 m/s, 33% localization errors are reduced for all the values of BER as shown in Fig. 17a. Thus, the impact of higher BER can be controlled by increasing f_b through a suitable value of v.

It is further illustrated in Fig. 17b, in which the location error at BER = 0.8 is reduced approximately 100% by increasing the value of v from 1 to 4. Hence, TMSL inherits the solution for higher BER since the parameter v is generally kept higher to tackle the higher nodes mobility and no additional procedure is incorporated.

6 Conclusion

Location information in WSAN is, inherently, as important as the event readings. Actor nodes require the exact location of the sources in order to identify the location where an event originates and take certain action. Numerous localization algorithms have been proposed for WSN. However, the localization of sensors independent of actors is not suitable in WSAN because actors need sensors location with respect to them for precise action. We propose a precise Timing-based Mobile Sensor Localization (TMSL) algorithm that follows the discrete time intervals to estimate the location of sensor nodes. This allows to measure the propagation time of the beacons locally at the receiving nodes, and thereby, making it independent of the clocks of sending node. In TMSL operation, actors are considered as reference nodes and broadcast *reference beacons* according to the defined pattern of intervals. In particular, TMSL works efficiently in highly mobile scenarios according to the application-specific localization accuracy level. Simulation results show that TMSL achieves high localization accuracy and restricts the errors by adapting the protocol parameters according to the expected velocity of nodes and required accuracy level.

References

1. Akyildiz IF, Kasimoglu IH (2004) Wireless sensor and actor networks: research challenges. Ad Hoc Networks 2(4):351–367
2. Peng H, Chuanshan G, Min Y, Dilin M, Bo Y (2005) ECLS: an efficient cooperative localization scheme for wireless sensor and actor networks. In: Proc. 5th international conference on computer and information technology (CIT'05), pp 396–400
3. Liu C, Wu K, He T (2004) Sensor Localization with ring overlapping based on comparison of received sidnal strength. In: Proc. IEEE mobile ad hoc and sensor systems (MASS'04), pp 516–518
4. He T, Huang C, Blum BM, Stankovic JA, Abdelzaher T (2003) Range free localization schemes for large scale sensor networks. In: Proc. 9th MOBICOM'03, pp 81–95
5. Priyantha NB, Balakrishnan H, Demaine ED, Teller S (2005) Mobile-assisted localization in wireless sensor networks. In: Proc. INFOCOM'05, pp 172–183
6. Fang L, Du W, Ning P (2005) A Beacon-Less location discovery scheme for wireless sensor networks. In: Proc. INFOCOM'05, pp 161–171

7. Niu R, Varshney PK (2006) Target location estimation in sensor networks with quantized data. IEEE Trans Signal Process 54(12):4519–4528
8. Langendoen K, Reijers N (2003) Distributed localization in WSN: a quantitative comparison. Comput Networks 43(4):499–518
9. Sheng X, Hu YH (2003) Collaborative source localization in wireless sensor network system. In: Proc. GLOBECOM'03
10. Datta S, Klinowski C, Rudafshani M, Khaleque S (2006) Distributed localization in static and mobile sensor networks. In: Proc. IEEE international conference on wireless and mobile computing, networking and communications (WiMob'06), pp 69–76
11. Hu LX, Evans D (2004) Localization for mobile sensor networks. In: Proc. ACM MOBICOM, pp 45–57
12. Moses RL, Krishnamurthy D, Patterson R (2003) A self-localization method for wireless sensor networks. EURASIP J Appl Signal Process 2003(4):348–358
13. Sichitiu ML, Ramadurai, V (2004) Localization of wireless sensor networks with a mobile beacon. In: Proc. 1st IEEE MASS'04, pp 174–183
14. Bulusu N, Heidemann J, Estrin D (2000) Gps-less low cost outdoor location for very small devices. IEEE Commun Mag 7(5):28–34
15. Nasipuri A, Li K (2002) A directionality based location discovery scheme for wireless sensor networks. In: Proc. ACM WSNA'02, pp 105–111
16. Savvides A, Han CC, Srivastava MB (2001) Dynamic fine-grained localization in ad hoc networks of sensors. In: Proc. 7th ACM/IEEE MOBICOM'01, pp 166–179
17. Cheng X, Thaeler A, Xue G, Chen D (2004) TPS: a time-based positioning scheme for outdoor wireless sensor networks. In: Proc. IEEE INFOCOM'04, vol 4, pp 2685–2696
18. Lazos L, Poovendran R (2004) SeRLoc: secure range-independent localization for wireless sensor networks. In: Proc. 3rd ACM workshop on Wireless security, pp. 21–30
19. Niculescu D, Nath B (2003) DV based positioning in ad hoc networks. J Telecommun Syst 22(1):267–280
20. Günther A, Hoene C (2005) Measuring round trip times to determine the distance between WLAN nodes. In: Proc. international IFIP-TC6 networking conference, pp. 768–779
21. Lee K, Park J, Lee B (2006) Dynamic localization with hybrid trilateration for mobile robots in intelligent space. In: Proc. 7th international workshop on human-friendly welfare robotic systems
22. Sanchez D (2007) Secure, accurate and precise time synchronization for wireless sensor network. In: Proc. 3rd ACM Workshop on QoS and security for wireless and mobile networks, pp 105–112
23. Simeone O, Spagnolini U (2007) Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators. EURASIP J Wireless Commun Network 2007:1–13
24. Bulusu N, Estrin D, Girad L, Heidemann J (2001) Scalable coordination for WSN: self configuring location systems. In: Proc. ISCTA'01
25. Thomas F, Ros L (2005) Revisiting trilateration for robot localization. IEEE Trans Robot 1(1):93–101
26. Coope ID (2000) Reliable computation of the points of intersection of n spheres in R^n. Austral, New Zealand Ind Appl Math J, Pt C 42:461–477
27. Network Simulator ns-2 (2008) The VINT Project. http://www.isi.edu/nsnam/ns/