Attack-Resistant Location Estimation in Sensor Networks

Outline

- Attacks
- Attack-Resistant Minimum Mean Square Estimation
- Voting-Based Location Estimation
- Simulation Evaluation
- Conclusion and Future Work

Attacks

- An attacker may provide incorrect location references by replaying the beacon packets intercepted at different locations.
- An attacker may compromise a beacon node and distribute malicious location references by lying about the beacon node’s location or manipulating the beacon signals.

Two Attack-Resistant Location Estimation Techniques

- Attack-Resistant Minimum Mean Square Estimation
- Voting-Based Location Estimation

Attack-Resistant Minimum Mean Square Estimation

- Idea
- Procedure
- Indicator
- A Simple, Threshold-Based Method

Idea

- A location reference introduced by a malicious attack is usually “different” from benign ones.
- When there are redundant location references, there must be some “inconsistency” between the malicious location references and benign ones.
- So?

Procedure

- First estimate the sensor's location with the MMSE-based method;
- Then assess if the estimated location could be derived from a set of consistent location references.
- If yes, accept the estimated result; otherwise, identify and remove the most "inconsistent" location reference.
- Continue the above process until find a set of consistent location references or not possible to find such a set.

Indicator

- Mean square error ς^2 as an indicator of the degree of inconsistency
  - Given a set of location references {<x_1, y_1, δ_1>, …, <x_m, y_m, δ_m>} and an estimated location (x_0, y_0),

$$\varsigma^2 = \sum_{i=1}^{m} \frac{(\delta_i - \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2})^2}{m}$$

  - δ: distance measured from its beacon signal
- Intuitively, the more inconsistent a set of location references is, the greater the corresponding mean square error should be.

A Simple, Threshold-Based Method

- A set of location references

  $\{<x_1, y_1, \delta_1>, \ldots, <x_m, y_m, \delta_m>\}$ obtained at a sensor node is τ-consistent w.r.t. a MMSE-based if the method gives an estimated location (x_0, y_0) such that $\varsigma^2 \leqq \tau^2$

How to Determine τ

- Derive the distribution of the mean square error using the real location as the estimated location, and compare it with the distribution obtained through simulation when there are location estimation error.
- Use this information to help determine τ.

How to Determine τ (cont.)

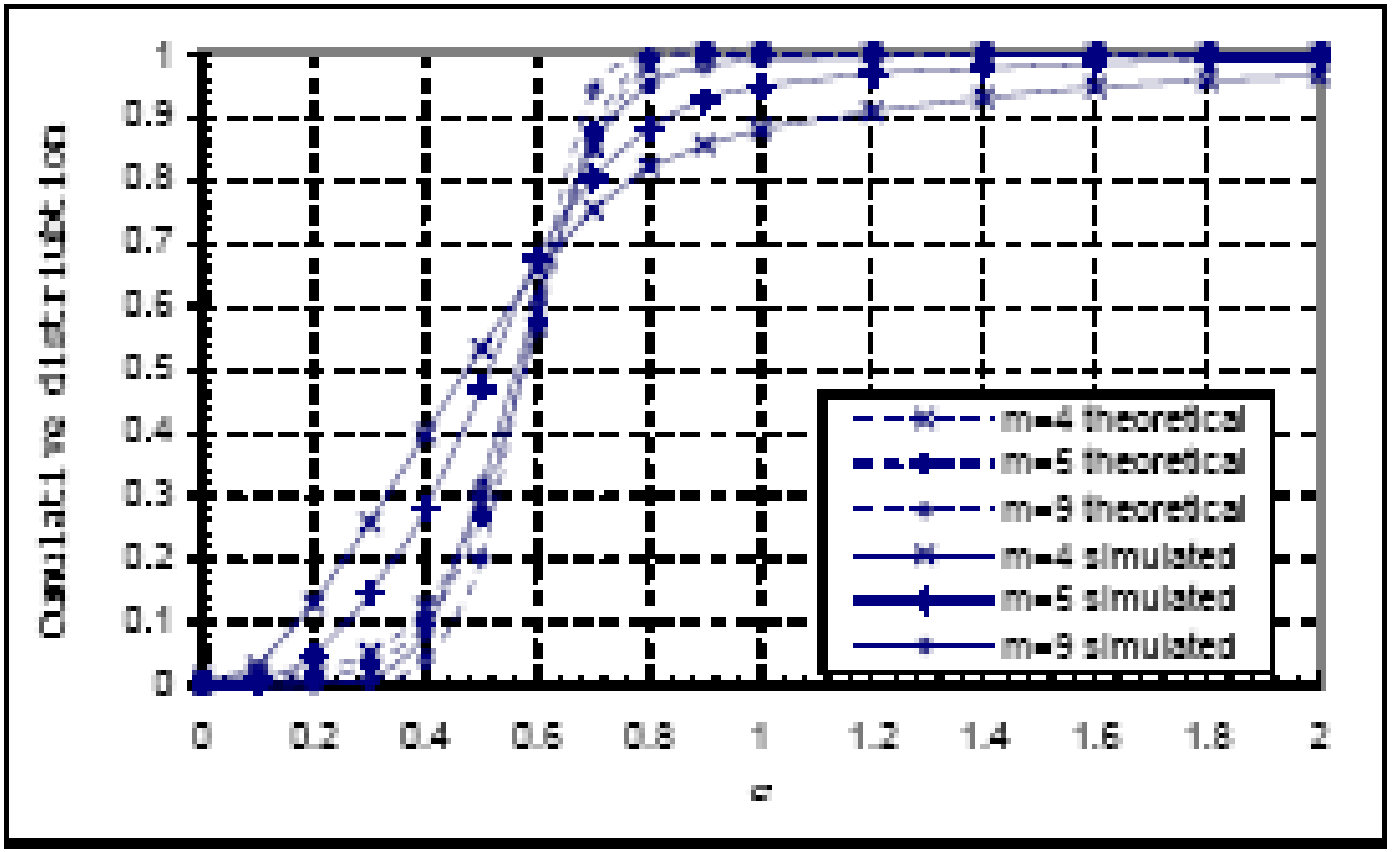


Fig. 3. Cumulative distribution function $F(\varsigma^2 \leq \varsigma_0^2)$. Let $c = \frac{\varsigma_0}{\sigma}$.

- When the number of location references m is large, the theoretical result is very close to the simulation results.
- However, when m is small, there are observable differences.

How to Determine τ (cont.)

- Choose the value for τ with a hybrid method
- When m is large, choose a value of τ corresponding to a high cumulative probability.
- When m is small, perform simulation to derive the actual distribution of the mean square error, and the determine the value of τ accordingly.

- Some discussion:
  - Is this approach really robust against a compromised anchor node?
  - Are they using the same key or personalized key?

Voting-Based Location Estimation

- The Basic Scheme
- Determine whether a ring overlaps with a cell
  - Overlap of a ring and a cell
- Iterative Refinement

The Basic Scheme

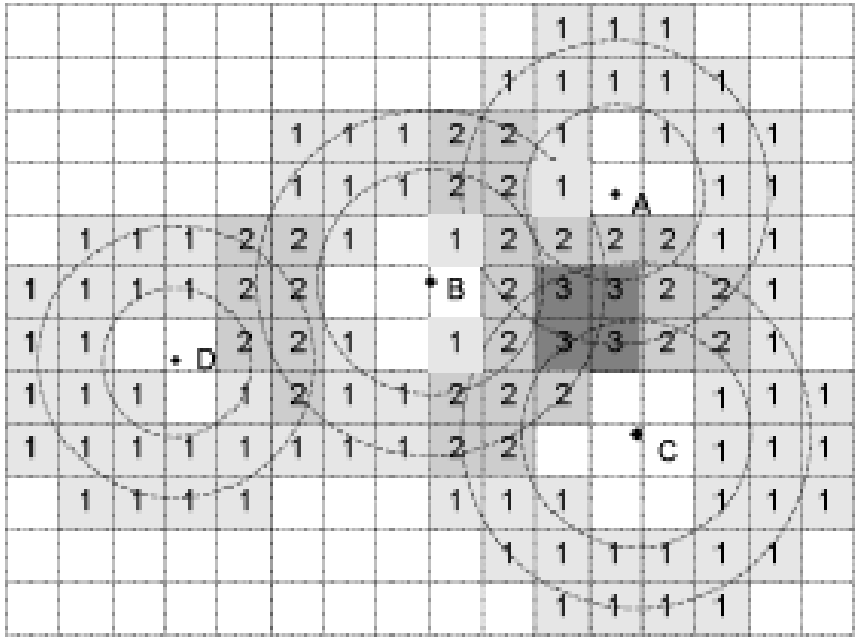


Fig. 4. The voting-based location estimation

- Identifies the minimum rectangle that covers all the locations declared in the location references and extends it by the max. transmission range of a beacon signal.
- Divides it into M small cells with the same side length L.
- Keeps a voting state variable for each cell, initially 0.

The Basic Scheme (cont.)

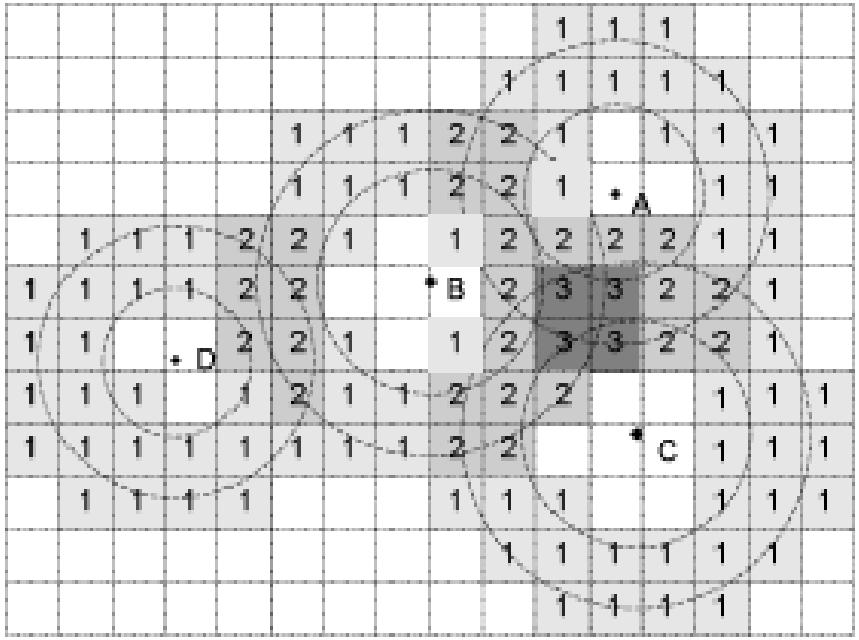
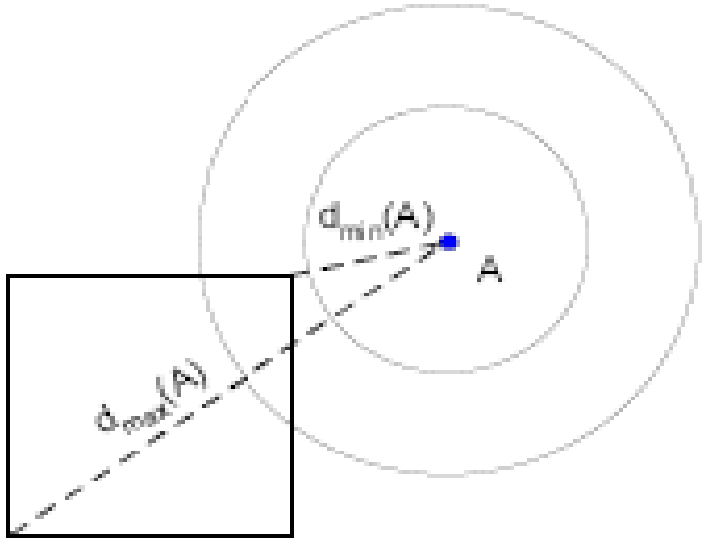


Fig. 4. The voting-based location estimation

- For each location reference <x, y, δ>, there is a candidate ring centered at (x, y) with inner radius max {δ-ε, 0}, the outer radius δ+ε.
  - δ: distance measured from its beacon signal
  - ε: maximum measurement error
- The sensor node identifies the cells that overlap with corresponding candidate ring, and increments the voting variables for these cells by 1.
- Finally chooses the cell(s) with the highest vote, and uses its (their) geometric centroid as the estimated location.

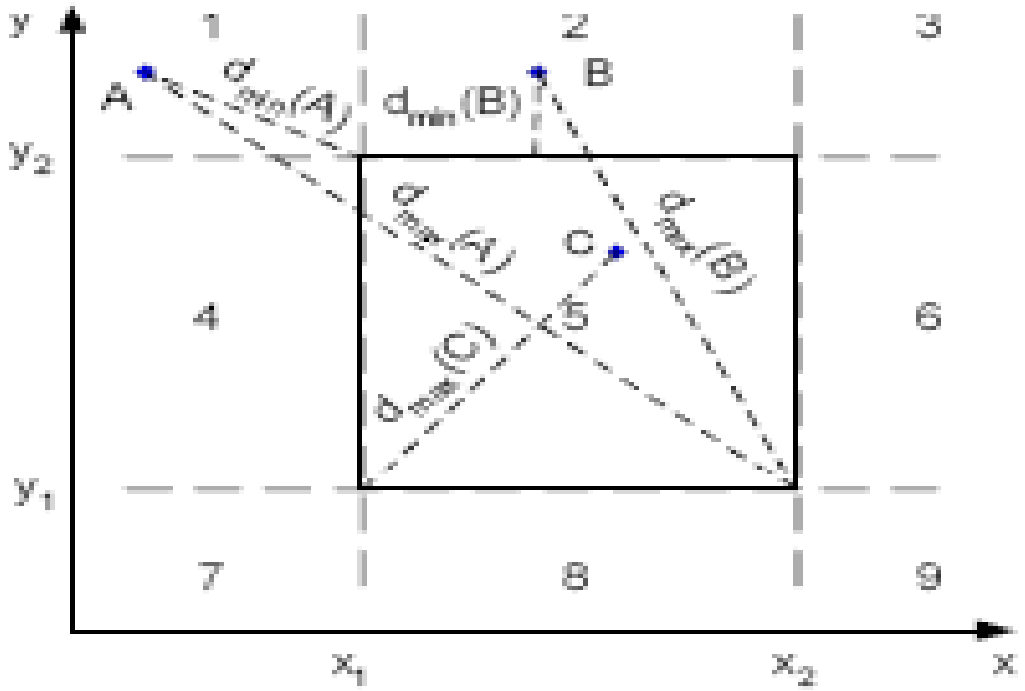
Overlap of Candidate Rings and Cells



(a) Overlap of a ring and a cell

- $d_{min}(A)$ and $d_{max}(A)$ denote the minimum and maximum distances from a point in the cell to point A.
- The candidate ring does not overlap with the cell when $d_{min}(A) > r_o$ or $d_{max}(A) < r_i$.
  - $r_i = \max\{0, \delta-\varepsilon\}$, $r_o = \delta+\varepsilon$

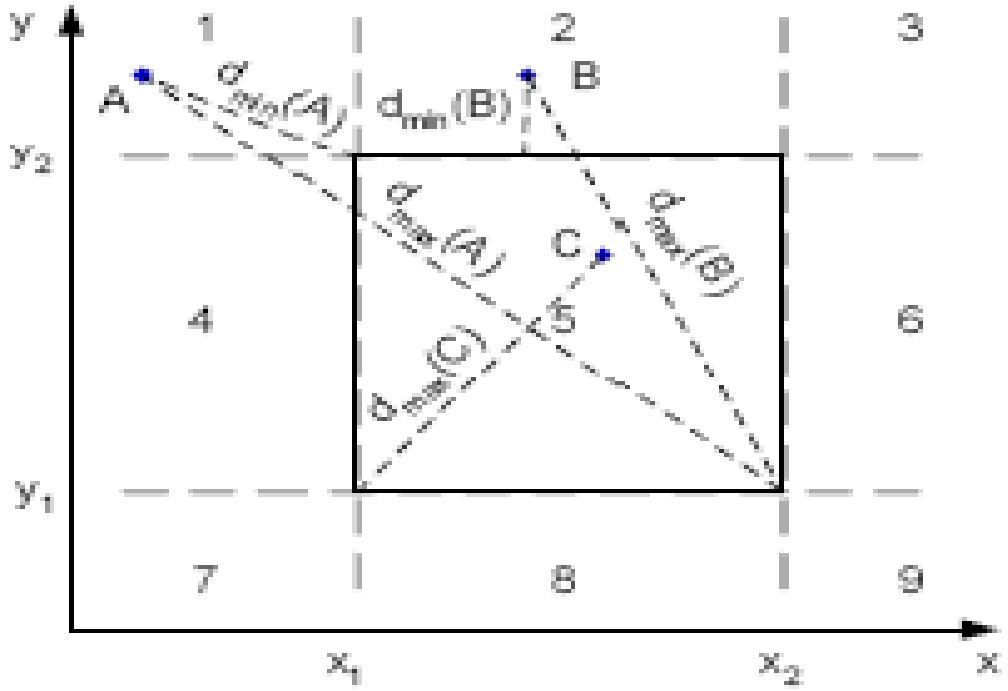
Computing d_{min} and $d_{max}(1)$



(b) Computing d_{min} and d_{max}

- $d_{min}(A)$ and $d_{max}(A)$ can be calculated according.
- Regions 3, 7, and 9 are similar.

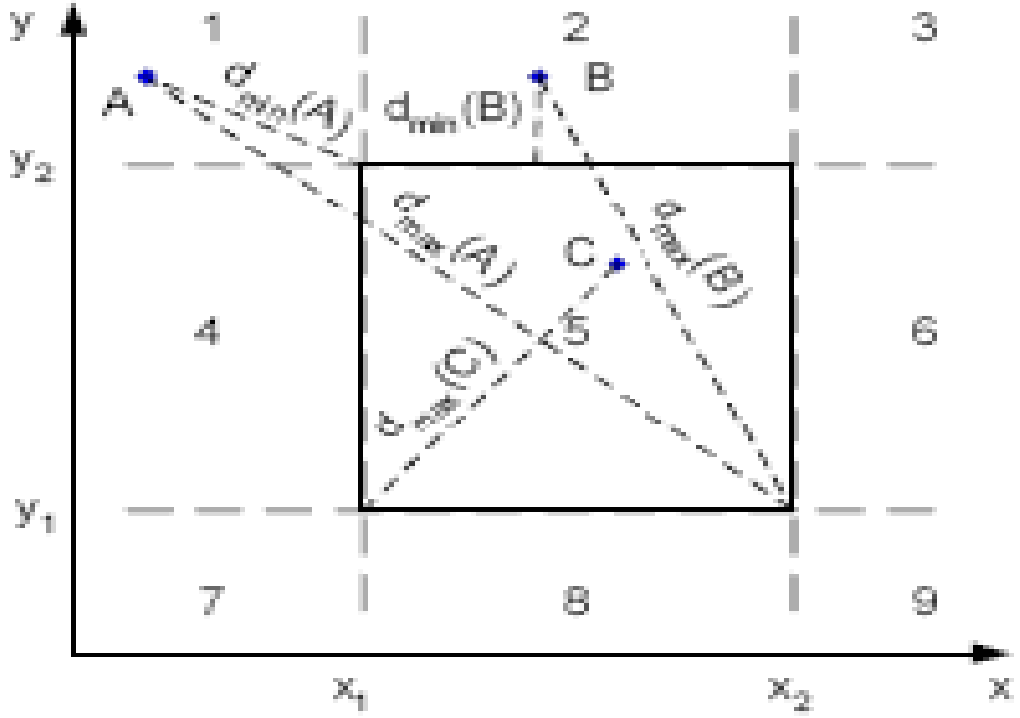
Computing d_{min} and d_{max}(2)



(b) Computing d_{min} and d_{max}

- $d_{min}(B) = y_B - y_2$
- If $x_B - x_1 > x_2 - x_B$? $d_{\max}(B) = \sqrt{(\max\{x_B - x_1, x_2 - x_B\})^2 + (y_B - y_1)^2}$
- Regions 4, 6, and 8 are similar.

Computing d_{min} and d_{max}(3)



(b) Computing d_{min} and d_{max}

- $d_{min}(C) = 0$, since point C is in the cell.
- Checks xc – x1 > x2 – xc and yc – y1 > y2 – yc,

$$d_{\max}(C) = \sqrt{(\max\{x_c - x_1, x_2 - x_c\})^2 + (\max\{y_c - y_1, y_2 - y_c\})^2}$$

Iterative Refinement (cont.)

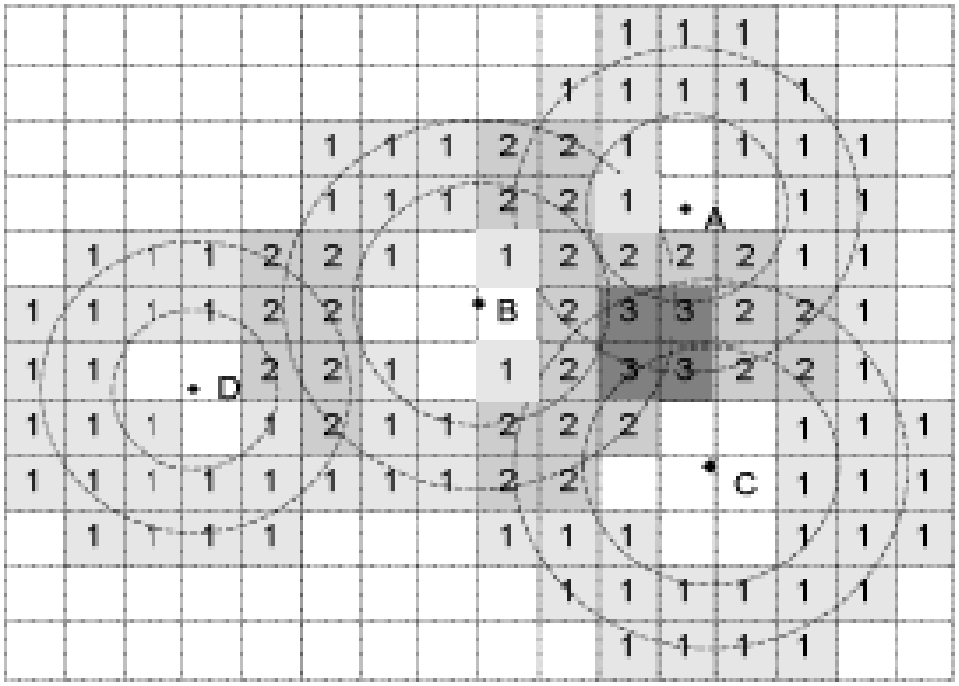


Fig. 4. The voting-based location estimation

- The number of cells M is chosen according to the memory constraint in a sensor node.
- After the first round of the algorithm, the node identifies the smallest rectangle that contains all the cells having the largest vote.
- A malicious location reference will be discarded (e.g., point D)

Simulation Evaluation

- Three Attack Scenarios
- Evaluation of Attack-Resistant MMSE
- Evaluation of Voting-Based Scheme

Three Attack Scenarios

- A single malicious location reference declares a wrong location.
- There are multiple non-colluding malicious location references, and each of them independently declares a wrong location.
- Multiple colluding malicious location references declare false locations that may appear to be consistent to a victim node.

Conclusion and Future Work

- An attack-resistant MMSE-based location estimation and a voting-based location estimation can deal with attacks in localization schemes.
- Study how to combine the proposed techniques with other protection mechanisms.
- Study the performance in a large scale.