Joint and Simultaneous K-sensor Detection in Deterministic and Random Sensor Networks

Yun Wang, Andrew Kutta
Department of Computer Science
Southern Illinois University Edwardsville
Illinois, USA
Email: yuwang@siue.edu, akutta@siue.edu

***Abstract*—**

***Keywords*-Intrusion detection; Intrusion path; Network deployment; Sensing range; Wireless sensor networks;**

I. Introduction

Wireless Sensor Networks (WSNs) are deployed for a broad range of civil or military applications to serve various purposes under different requirements. As an example, some application requires an object being detected by at least k sensors *simultaneously* to fulfill its task such as determining the object's location and track its trajectory. This **simultaneous-k-sensing-detection** is of paramount importance to some WSN applications [reference]. Fig. 1 illustrates an instance of *simultaneous* K-sensor detection of an object. In the depicted scenario, sensors s_1, s_2, s_3 and s_4 can simultaneously detect the object but only s_1, s_2 and s_3 can send the sensing data to the base station(BS) as they are connected with the BS while sensor s_4's detection does not contribute to the simultaneous-k-sensing-detection as it is isolated from the BS. In other words, the scenario shown in Fig. 1 only satisfies the requirements of simultaneous-3-sensing-detection.

In fact, k-sensing detection is not a brand new concept and has been discussed extensively in the literature. However, most of the research work are focus on **joint-k-sensing-detection**, in which an mobile object can be detected by the WSN as long as at least k sensors can detect it within application-specified time or distance. Fig. 2 illustrates an instance of joint-k-sensing-detection, in which s_1, s_2, and s_3 can not detect the mobile object simultaneously, but they can jointly detect it before the object travels a maximum allowable intrusion distance of D_{max} in the WSN.

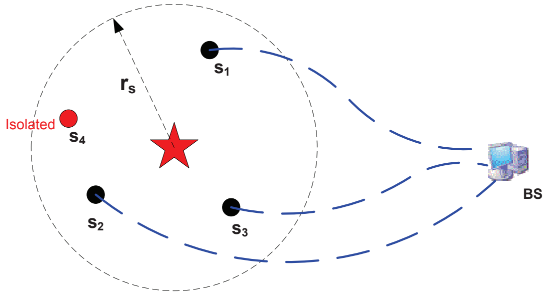


Figure 1. An instance of Simultaneous K-sensor detection of a static/mobile object

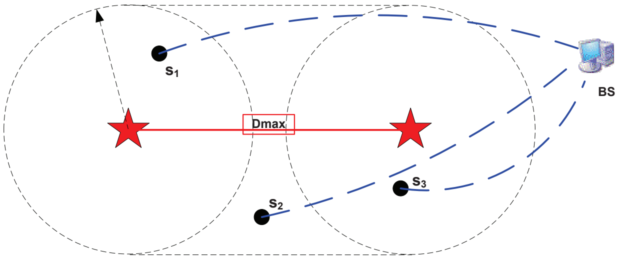


Figure 2. An instance of Jointly K-sensor detection of a mobile object

To be completed by YW

II. Related Work

III. Intrusion Detection Network Model and Definitions

Our network model includes a network model, a intrusion detection model, and the performance evaluation metrics.

A. Network Model

We consider a homogeneous WSN in a two dimensional square plane with an area of $A = L \times L$. A number of N sensors, denoted by a set $\mathbf{N} = (n_1, n_2, \ldots, n_N)$, are deployed in A following deterministic or random sensor deployment approach. In random deployment, sensor are assumed to be uniformly and independently following a two-dimensional Poisson point distribution, and in random deployment, sensors are deployed following a square or triangle pattern, as illustrated in Fig. 3. In such a network model, all sensors are assumed to be static once the WSN has been deployed and all have the same sensing range of r_s and a transmission range of r_c.

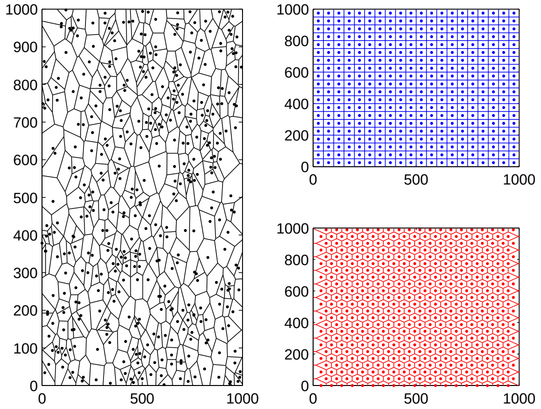


Figure 3. Random and Deterministic Sensor Deployment

B. Sensing and Detection Model

- **simultaneous-k-sensing-detection**: To be completed by YW
- **Joint-k-sensing-detection**: To be completed by YW

C. Evaluation Metrics

In order to evaluate the quality of intrusion detection in WSNs, we use the metrics defined in [1] as follows:

- **Intrusion Distance**: The intrusion distance, denoted by D, is the distance between the point where the intruder enters the WSN and the point where the intruder gets detected by any sensor(s). ξ is the maximal distance allowable for the intruder to move before it is detected by the WSN for an application.
- **Detection Probability**: The detection probability is defined as the probability that an intruder is detected within the Maximal Intrusion Distance, ξ)
- **Average Intrusion Distance**: The average intrusion distance is defined as the expected distance that the intruder travels before it is detected by the WSN for the first time.

IV. Analysis and Derivation

To be completed by YW.

V. Simulations

A. Design and Implementation

Implemented are two different interfaces, a Graphical User Interface and an Automated Driver. The User Interface uses a single thread to run a single iteration of the simulator with the user defined parameters. The Automated Driver uses three threads to simulate Simultaneous K Detection, Joint K Detection, and Single Detection Sensing Models. Each detection type is tested with variable sensing and communication ranges averaged over a user defined number of iterations per range before incrementing either the communication or sensing range.

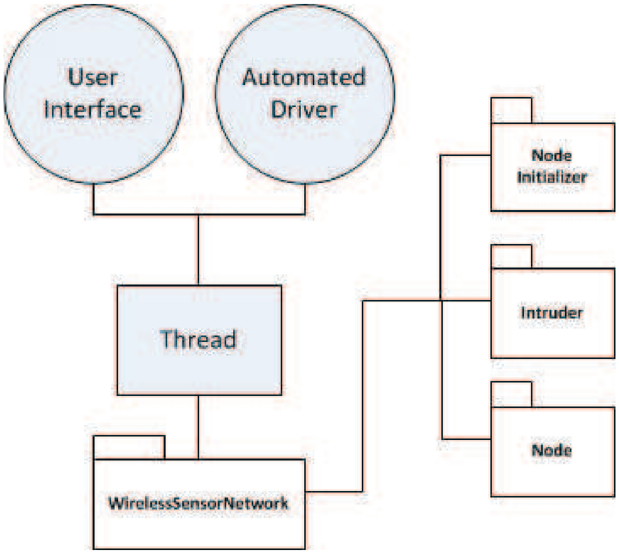


Figure 4. Generalized Architecture of the Simulator

The WSN simulator includes three packages: Node, Intruder, and Node Initializer. The Node Initializer is used to place the Nodes according to the selected distribution pattern such as random, square, or triangle. The Intruder is called by the WSN to update it's position at every tick, once updated the WSN package iterates through all Nodes to determine if the Intruder is within the sensing range. A sub-package of Node is the Base Station (BS). The BS handles all detections and filters out invalid detections due to not meeting detection type or due to lack of connectivity.

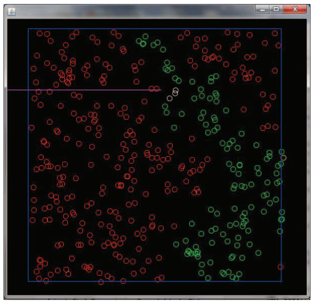
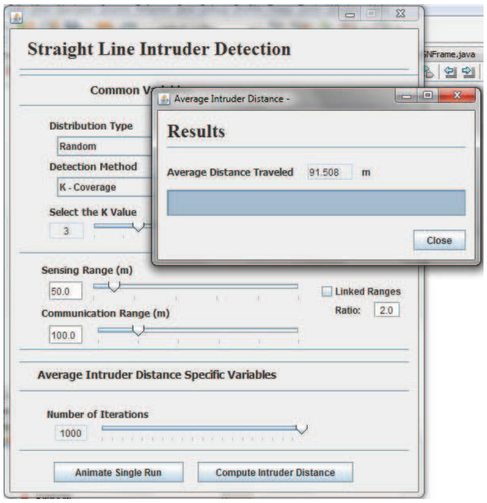


Figure 5. Example Iteration with Communication Range equal to 70

The Graphical User Interface can be used to demonstrate a single iteration with the user defined parameters, or run a variable amount of iterations with the parameters. As illustrated in 5, with random distribution it is possible to have many of the nodes that would normally be fully connected be sectioned off and non-viable for detecting intruders.

The second functionality of the GUI is to compute the average distance traveled by an intruder. This interface is shown in Fig. 6. It roots more into the automated driver method by creating a thread and performing up to 1000 iterations.



Straight Line Intruder Detection

Common V

Distribution Type: Random

Detection Method: K - Coverage

Select the K Value: 3

Sensing Range (m): 50.0

Communication Range (m): 100.0

☐ Linked Ranges

Ratio: 2.0

Average Intruder Distance Specific Variables

Number of Iterations: 1000

Average Intruder Distance -

Results

Average Distance Traveled: 91.508 m

Figure 6. Interface when computing average intruder distance over 1000 iterations

B. Results and Discussions

For every Sensor Distribution Models used, the sensing range begins at 27 meters and is incremented by 2 meters until a maximum of 77 meters is reached. For each sensing range, the test was averaged over 1000 iterations in order to achieve a reliable result.

In order to determine whether the intruder is detected, three detection strategies are deployed. These strategies will be explained later in this section. Each strategy relies on being connected to a base station. The sensor network is deployed over a 1000x1000 meter area with the base station statically located at 1000x500 in each deployment method. General strategy used to determine connectivity is the Dijkstra's Algorithm. The version implemented is explained below.

The Sensor Connectivity graph is the result of running Dijkstra's Algorithm after each generation of the Sensor Network. A generation of the Sensor Network is considered a single iteration. There are 1000 iterations performed per sensing range, or, in this case, communication range. The communication range is defined as twice the sensing range. It is easily seen that the limiting factor for the random distribution is the connectivity of the network. Although, initially the random distribution performs better.

Algorithm 1 Dijkstra's Algorithm

```
Require: $nodes.length() \geq 1$
  $openNodeList \Leftarrow newList$
  $openNodeList.add(0)$
  while $openNodeList.size() \neq 0$ do
    $curNode \Leftarrow ClosestNode$
    for $i = 1$ to $nodes.length()$ do
      if $nodes[i] \neq Finalized$ then
        if $NodesWithinRange(curNode, i) = $ true then
          if $openNodeList.contains(i) \neq$ true then
            $nodes[i].Head \Leftarrow curNode$
            $openNodeList.add(i)$
          end if
        end if
      end if
    end for
    $nodes[curNode] \Leftarrow Finalized$
    $openNodeList.remove(curNode)$
  end while
```

In the simulator created, the universal update() call is sent to the Wireless Sensor Network, which in turn requests that the intruder updates as well. When an intruder is updated, it's position is updated based on it's own algorithm. For the Straight-Line Intruder, this means that the intruder moves one meter towards the goal. After the intruder is updated, the wireless sensor network must check for a detection. This is done using the below algorithms where IntruderWithinRange is defined as the Euclidean Distance of the intruder compared with the sensor. The distance is then compared with the sensor's sensing range.

At this point, if the sensor has detected an intruder, it sends that information to the base station. The information transfered to the base is mainly the index of the current sensor within the array.

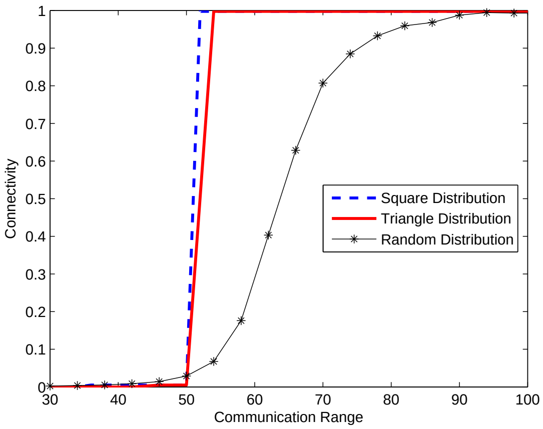


Figure 7. Connectivity of Sensors. Communication range of sensors are 2∗ Sensing Range

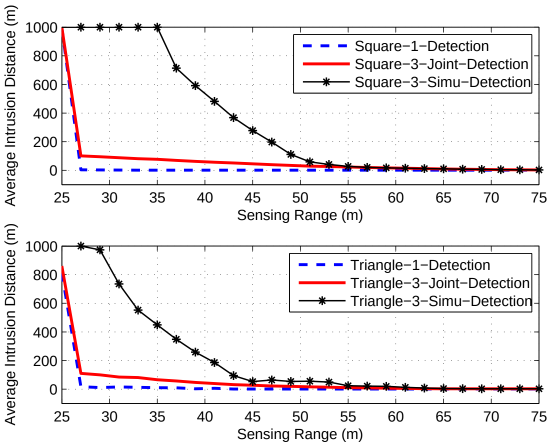


Figure 8. Average Intrusion Distance in a lattice WSN following Square and Triangle Patterns

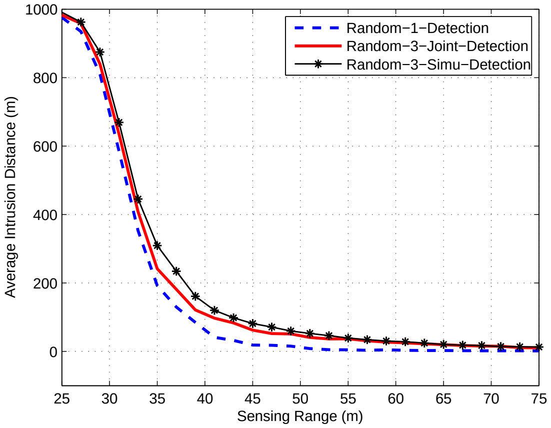


Figure 9. Average Intrusion Distance in a Random WSN

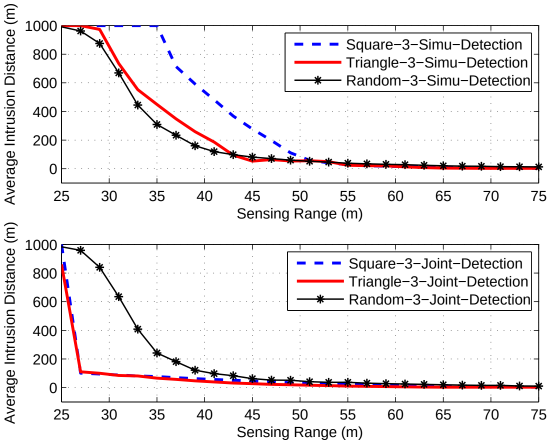


Figure 10. Average Intrusion Distance in a Random WSN under 3 jointly and Simultaneously Detection

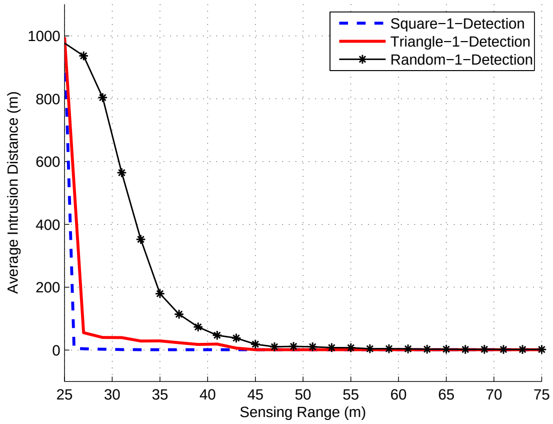


Figure 11. Average Intrusion Distance in a Random WSN under single-sensing detection

Algorithm 2 Check For Detection

```
for $i = 1$ to $nodes.length()$ do
    if $IntruderWithinRange$ then
        $sendToBase()$
    end if
end for
```

Once every node has been checked for a detected intruder, the base station processes all of the potential detections. At this point the Base Recieve Detection algorithm is employed. Before any processing occurs, it checks to make sure that the detection is valid. It is only considered valid if the node is able to communicate with the base station. If the sensor cannot communicate with the base, then the detection message is ignored. Otherwise the base station counts the detection and raises the Detected flag depending on the sensing strategy being used.

C. Single-sensing Detection

The Single-Sense Detection Strategy will flag the intruder as detected by any node that can communicate with the Base Station which is located at $(1000, 500)$ in the Sensor

Algorithm 3 Base Recieve Detection

```
Require: $node \neq null \vee curNode \geq 0$
    if $node.distanceToBase > 0$ then
        if $SensingType = SINGLEDETECTION$ then
            $Detected = true$
        else
            $nodesWithDetect[curNode] \Leftarrow node$
            $curNode \Leftarrow curNode + 1$
        end if
    end if
```

Network. As can be seen, as soon as the Square and Triangle Distributions become fully connected, the random distribution is easily out performed.

D. *K-sensing Joint Detection*

In the simulations run, K is set to 3. A 3-Sensing Single Detection strategy is when three sensors detect the intruder independantly. Each detecting sensor can be completely independant of each other. In essence, this strategy is similar to that of the Single Sensing strategy, but requires a minimum number of nodes to flag the intruder as detected.

In comparison with the Single Detection strategy, the randomness of the non-random distribution strategies are removed as can seen with a sensing distance of less than 25 meters. However, once fully connected the uniformly distributed strategies easily out perform the random distributions.

E. *K-Simultaneous Detection*

In the K coverage sensing model, with the simulation run with $K = 3$, the detection is only flagged if there are 3 sensors that detect the intruder at the same frame in time. In other words, the intruder is considered detected if 3 sensors detect the intruder at a single location. This is the most interesting case of sensing in that initially the random distribution model becomes much more effecient. Once the square and triangle distributions become fully connected at 26 meters, the sensors are not able to create a reliable K Coverage sensor coverage until the sensing range becomes 50 meters. The reason this occurs is that even though the sensor network is fully connected, there is a possibility that the intruder decides to travel directly along a line of sensors resulting in at most 2 simultaneous detections. The uniform distributions become more effecient than the randomly distribution at this point.

VI. Analysis and Simulation Validation

We execute the simulations in a homogeneous WSN scenario with 400 sensors uniformly deployed in a $1000 * 1000$ square meters deployment field. The sensing range of each sensor varies from 0 to 100 meters. Monte Carlo simulations are performed. All results shown here are the average of 1000 simulation runs.

A. *Effects of varying network parameters on intrusion detection probability*

In order to examine the effects of varying network parameters such as distribution type and sensing range on intrusion detection probability following different paths. Straight-line is defined as $y = r$ where r is randomly generated.

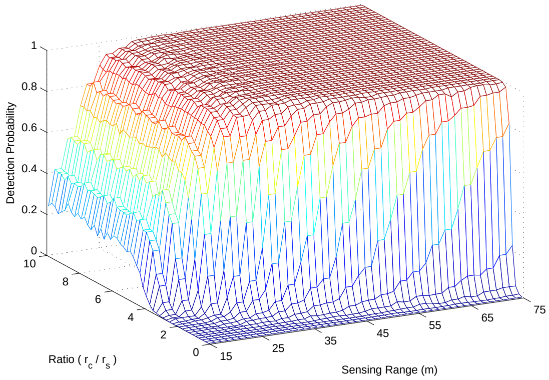


Figure 12. RandomKCover DetectionProb VariableRatio eps

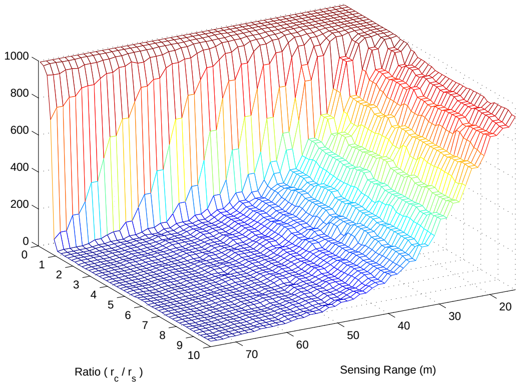


Figure 13. RandomKCover DistanceTraveled VariableRatio.eps

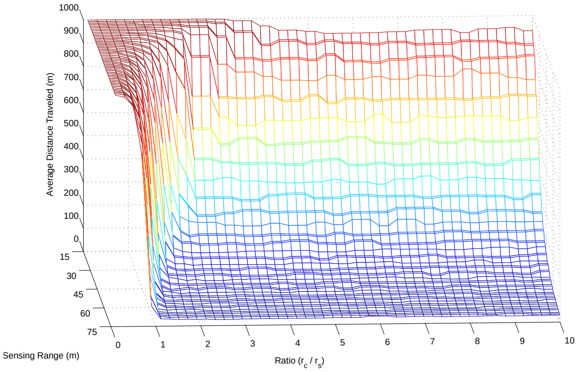


Figure 14. RandomKCover DistanceTraveled VariableRatio v2.eps

VII. Conclusion

References

[1] Y. Wang, X. Wang, B. Xie, D. Wang, and D. P. Agrawal, “Intrusion detection in homogeneous and heterogeneous wireless sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 7, no. 6, pp. 698–711, 2008.