Chapter 3

Analysis of k-Target Tracking and m-Connectivity in Directional WSNs

3.1 Introduction

A Wireless Sensor Network (WSN) consists of several sensor nodes for target tracking of objects in a Field of Interest (FoI). The literature on WSNs mostly assumes the omni-directional tracking model where a sensor node can perfectly track a target within the disk-shaped tracking region of the sensor node Wu et al. (2015). However, there are many sensor nodes in which the tracking region is limited to a fixed direction Yu et al. (2011); Zhao et al. (2013). Examples of such sensor nodes are camera sensors, multimedia sensors, infrared sensors, ultrasonic sensors, lidar sensors, and radar sensors. A directional sensor node captures more accurate and comprehensive information and consumes low power for target tracking, indoor navigation, positioning system, and surveillance applications Kumar and Sivalingam (2012).

Target tracking in a WSN is considered as an important parameter to measure the tracking quality of a Target Tracking System (TTS) Fanti et al. (2016); Gupta and Rao (2016); Lu et al. (2017); Mahboubi et al. (2017); Xiong et al. (2017); Zheng et al. (2016).

A path travelled by a target in the FoI can be divided into segments. The length of a segment of a path is the Euclidean distance between the starting and ending points of the segment. A moving target in the FoI is said to be *k-target tracked* if at least k directional sensor nodes are simultaneously tracked the target in a given segment length. Fig. 3.1 illustrates a TTS where the sensor nodes are placed in a square pattern and a target is moving in the FoI. A directional sensor node transmits the tracking information to the sink when the sensor node tracks a moving target. Data communication from a directional sensor node to the sink is an important matrix to measure the connectivity probability of a TTS. A directional sensor node in a WSN is m-connected if there are m connected paths from the sensor node to the sink in the network, where $m \geq 1$.

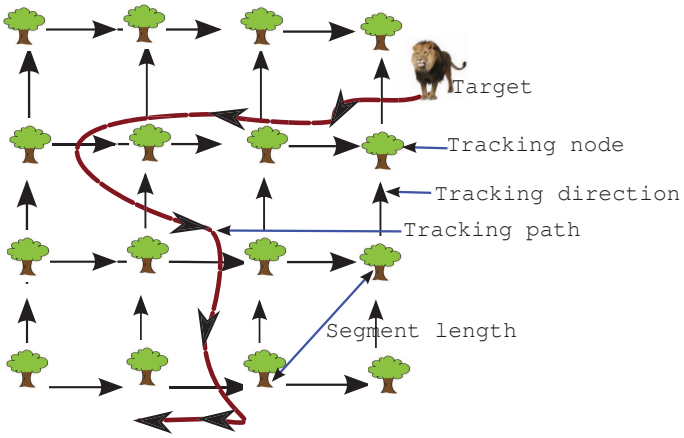

Figure 3.1 Illustration of a Target Tracking System (TTS).

In this chapter, we improve the quality of the TTS in WSNs by employing optimal sensor node placement strategy. We consider equilateral triangle, square, and hexagon regular sensor node deployment patterns. *We address the problem: What is the regular sensor node deployment pattern that uses the minimum number of sensor nodes for k-target tracking in m-connected WSNs, where $k \geq 1$, $m \geq 1$, and segment length are given?*

3.1.1 Contributions of Work

To the best of our knowledge, this is the first work to address the target tracking problem using directional sensor nodes, *i.e.*, to determine the location of the sensor nodes, distance between sensor nodes, and direction of the tracking region of the sensor nodes for the desired level of target tracking, network connectivity, and the segment length. This work helps to develop a minimum cost TTS for the designed level of tracking and connectivity. Apart from this, the contributions of this research work are as follows:

- We solve a regular directional sensor node placement problem where a moving target is tracked by at least k sensor nodes and each sensor node in the WSN is m-connected. The objective of the problem is to determine the direction of the tracking region and location of the sensor nodes for the given target tracking and network connectivity. The solution helps to calculate the required number of directional sensor nodes for a given regular deployment pattern. The solution finally selects a deployment pattern which requires the least number of sensor nodes for the designed level of tracking and connectivity.

- It is difficult to calculate the location of directional sensor nodes in an irregular FoI. We propose an algorithm to compute the location of the sensor nodes in the FoI. The correctness of the algorithm is validated by numerical and simulation results. The results illustrate that only one regular directional sensor node placement pattern is suitable for tracking a target by k sensor nodes in a connected WSN under different values of the parameters of a TTS.

- We develop a TTS using the proposed analysis for tracking the moving targets in the FoI. We setup a testbed with different Scenarios for conducting experiments. Target tracking data are collected to evaluate the proposed work. The experiment results validate the solution of the target tracking problem and demonstrate the impact of the parameters of the TTS on different sensor node placement patterns.

3.1.2 Motivation and Limitations in Existing Work:

The work in this chapter is motivated by the following shortcomings of existing work.

• **1-target tracking:** The existing work for solving the target tracking problem considers only 1-target tracking Bai et al. (2006); Derr and Manic (2015); Kumar and Sivalingam (2012); Namboothiri et al. (2011); Seok et al. (2013); Yun et al. (2012, 2010). Moreover, some existing work on sensor node placement for target tracking in WSNs proposed the solutions only for a selected region, where the FoI is partially tracked Liao et al. (2011); Xu et al. (2014). However, 1-target tracking or partial tracking is not suitable in some applications of the WSNs such as indoor navigation, positioning system, and surveillance applications. Such applications require that the target is simultaneously tracked by at least three sensor nodes.

• **No network connectivity:** The work in Birjandi et al. (2013); Derr and Manic (2015); Kim et al. (2009); Seok et al. (2013) considers only target tracking problem in the FoI. Connectivity between directional sensor nodes is also an important metric in WSNs. The authors in Yun et al. (2012) show that separate solutions of a target tracking problem and connectivity problem in WSNs do not provide a minimum cost WSNs.

• **Omnidirectional target region:** The work in Bai et al. (2010); Gupta et al. (2015b); Yun et al. (2012, 2010) assumed that the tracking region of the sensor nodes are omnidirectional for solving the problem in WSNs. The authors in Kranakis et al. (2005) proved that the omnidirectional tracking region consumes more energy than the directional tracking region and creates an interference problem.

• **Minimum cost WSNs:** The existing work Kumar and Sivalingam (2012) considered only square-grid deployment pattern of sensor nodes for solving target tracking problem in a directional WSN. However, triangle and hexagon deployment patterns may require less number of sensor nodes for the designed level of target tracking and connectivity of WSNs.

• **Limitation of simulation results:** The existing work used only simulation results

for validating the work Bai et al. (2010); Gupta et al. (2015b); Yun et al. (2012, 2010). However, simulation study does not cover the practical challenges. Examples of such challenges are infeasible location of sensor node deployment and fixed boundary of the deployed region. Moreover, the simulation study assumes that the generated results are correct and does not use any other cross validation method (*e.g.*, camera based or human monitoring) to verify the results.

Table 3.1 illustrates the comparative summary of the existing work. It shows that the none of the existing work consider k-target tracking and m-connectivity for identify the optimal deployment pattern (triangle, square, or hexagon patterns), critical sensor density, and the position of the sensor nodes in a directional WSNs.

Table 3.1 A comparative summary of the existing work. The primary objects are Optimal Deployment Pattern (ODP), Critical Sensor Density (CSD), and Position of Sensor Nodes (PoN).

Paper	Antenna	k-tracking	m-conn.	Primary object
Yu et al. (2011)	Directional	1	1	ODP
Kumar and Sivalingam (2012)	Directional	1	1	PoN
Yun et al. (2012)	Omni	1	$(3, .., 6)$	ODP
Yun et al. (2010)	Omni	1	≤ 6	ODP
Bai et al. (2010)	Omni	1	4	ODP and PoN
Liao et al. (2011)	Omni	1	1	ODP
Seok et al. (2013)	Omni	1	$\times$	ODP
Bai et al. (2006)	Omni	1	$1, 2$	ODP
Derr and Manic (2015)	Omni	1	$\times$	PoN
Birjandi et al. (2013)	Omni	k	$\times$	ODP and PoN
Kim et al. (2009)	Omni	1	$\times$	ODP and PoN
hwan Kim et al. (2012)	Omni	k	≤ 6	ODP
Wu and Wang (2012)	Directional	1	$\times$	CSD
Khanjary et al. (2014)	Directional	1	1	CSD
Ma et al. (2009)	Directional	1	1	ODP
Wang and Zhang (2014)	Directional	1	1	CSD
Khan et al. (2017)	Directional	1	$\times$	PoN
Dou and Nan (2017)	Omni	1	$\times$	CSD
Zhu et al. (2018)	Directional	1	1	CSD

Considering these limitations in the existing work, we determine the location of the sensor nodes, distance between sensor nodes, and direction of the tracking region of the sensor nodes for tracking a moving target by at least k directional sensor nodes and m-connected WSNs, where $k \geq 1$ and $m \geq 1$. We also develop a TTS using the proposed analysis for tracking the moving targets in the FoI. The novel features of the propose solution for

solving the target tracking problem are as follows:

• Different from the existing work, the propose analysis consists of the segment length and geometry of the regular deployment patterns (square, triangle, and hexagon) for solving the problem. The novelty of the analysis is to develop the correlation between the distance of the sensor nodes and the direction of the tracking region. Such correlation provides the designed level of target tracking in a connected WSN using directional sensor nodes. The analysis helps to design a minimum cost m-connected WSN for k-target tracking. The directional sensor node also consumes less energy and reduces an interference problem than omnidirectional sensor nodes.

• The next novel feature of the proposed system is to handle the practical challenges, *i.e.*, infeasible location of the sensor node. Such challenges are possible if the estimated location of the sensor node is outside the FoI but the tracking region of a sensor node is partially required as shown in Fig. 3.7(a). The propose system deploys the sensor node on the boundary of the FoI so that the sensor node tracks the moving target in the given direction as shown in Fig. 3.7(b).

• Finally, the propose system is validated by using prototype results. Different from the exiting work, the prototype uses the proposed WSNs and camera for cross validation. Such cross validation helps to identify the correlation of the proposed work.

The rest of the chapter is organised as follows: The next section states the terms, notations, and assumptions and formulate the target tracking problem. The solution of the problem is proposed in Section 3.3. Numerical and simulation results are presented in Section 3.4. Section 3.5 considers indoor and outdoor sensor node placement Scenarios. It presents prototype results to validate and demonstrate the impact of the parameters on the TTS. Section 3.6 concludes the chapter.

3.2 Preliminaries and Problem Definition

This section introduces the assumptions, tracking model, and communication model. We also define the target tracking problem.

3.2.1 Assumptions

We assume that the directional sensor nodes are deterministically placed in a FoI Ψ. We consider three possible regular placement patterns of the sensor nodes for solving the target tracking problem. A regular pattern can be constructed by using equilateral triangle, square, or hexagon shaped regular polygon as shown in Fig. 3.2. The side length of a polygon is denoted by l, where $l > 0$. In this work, a polygon is denoted by p, where the value of p equals to 0, 1, and 2 for triangle, square, and hexagon patterns, respectively. The position of a sensor node in the FoI is denoted by (x, y), where $(x, y) \in \Psi$, $x \geq 0$, and $y \geq 0$.

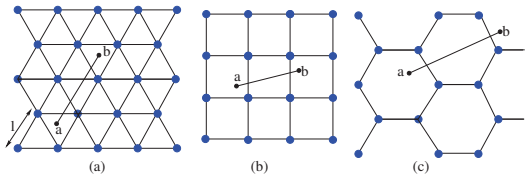

Figure 3.2 Regular directional sensor node placement patterns in the FoI. Parts (a), (b), and (c) illustrate the equilateral triangle, square, and hexagon patterns, respectively. Point **a** and point **b** show the starting and ending points of a segment, respectively.

3.2.2 Tracking Model

In this chapter, we consider directional tracking model that can yield very accurate results for monitoring or tracking applications. This section discusses the main characteristics of a directional sensor node, which directly affects the accuracy of target tracking. Fig. 3.3 shows a directional sensor node.

- **Tracking angle:** Directional sensor nodes have a limited tracking angle, denoted by ϕ, due to the hardware constraints and cost considerations, $1 \leq \phi \leq 360°$. In this work, we assume a small value of tracking angle.

- **Tracking range:** Tracking range is defined as the maximum range of a sensor node within which the sensor node can track a target in the FoI. In a WSN, the tracking range of a sensor node is denoted by R_s, where $R_s > 0$.

- **Orientation of the tracking region:** It is defined as the orientation to which a directional sensor node point in the FoI. Sensor nodes in WSNs may have different orientation based on the deterministically placement of the sensor nodes. In this work, the orientation vector of a sensor node s located at (x, y) is denoted by $\theta_{x,y}$, where $0 \leq \theta_{x,y} \leq 2\pi$.

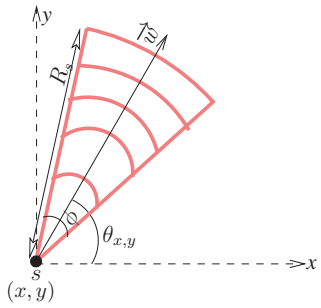

Figure 3.3 Illustration of a directional sensor node.

3.2.3 Communication Model

In this work, we use omnidirectional communication model where a sensor node s can communicate with other sensor nodes within the disk of a given communication radius, denoted by R_c, centered at the location of the sensor node s. The communication region is denoted by $A(s, R_c)$. We assume that all sensor nodes in a WSN have equal communication range.

Definition 3 *A sensor node s in the WSN is said to be a neighbor of another sensor node z if the Euclidean distance between the location of the sensor nodes is less than the communication range of the sensor nodes. In m-**connected** WSNs, the number of neighboring sensor nodes of z is equal or more than m, where* $m \geq 1$.

Definition 4 *Let a and b are the starting and ending points of a segment in a path, respectively, travelled by a target in the FoI. The length of the segment is the Euclidean distance between the points a and b as shown in Fig 3.2. Let l be the length of an edge in the regular patterns. Fig. 3.2 illustrates that a target crosses the tracking region of at least one sensor node if*

$$\textit{Segment length} \geqslant \begin{cases} l & \textit{if } p{=}0 \\ \sqrt{2}l & \textit{if } p{=}1 \\ \sqrt{3}l & \textit{if } p{=}2 \end{cases}$$

$$\textit{Segment length} \geqslant l\sqrt{p+1}. \tag{3.1}$$

*A target in the FoI is said to be k-**target track** if the target crosses the tracking region of at least k sensor nodes simultaneously in a given segment length, where segment length* $\geq l\sqrt{p+1}$, $k \geq 1$, *and* $p = \{0, 1, 2\}$. *Fig. 3.5 illustrates the Scenarios of* 1*-target tracking and* 2*-target tracking.*

Definition 5 *In a given segment length, a regular pattern is known as **optimal pattern** if the pattern provides k-target tracking and network is m-connected with minimum required number of directional sensor nodes.*

3.2.4 Target Tracking Problem

Given a FoI Ψ, R_s, R_c, k, m, and segment length. The target tracking problem is to determine the location and orientation of tracking region of the directional sensor nodes in the FoI Ψ, such that

- *C1:* The moving target is tracked by at least k directional sensor nodes
- *C2:* Sensor nodes in the network are m-connected
- *C3:* Target must be tracked in a given segment length
- *C4:* Size of the network is minimum.

Here, conditions *C1*, *C2*, and *C3* require for k-target tracking in m-connected WSNs and *C4* provides a minimum cost WSN for TTS.
In this problem, we have to ensure that a target in the FoI is tracked by at least k sensor nodes simultaneously within the segment length and each sensor node in the WSN is connected with m sensor nodes, where $\{k, m\} > 0$. The target tracking problem can be solved by converting into regular placement patterns.

3.3 Solution of Target Tracking Problem

In this section, we propose a solution for solving the target tracking problem. The solution consists of the following four steps. Step 1 and Step 2 estimate the coordinates and orientation of the tracking region of the sensor nodes, respectively. Step 3 estimates the optimal side length in a polygon of a given pattern for desired target tracking in a connected WSN. Such optimal side length is the maximum possible distance between two nodes in a polygon. The optimal side length and regular pattern help to calculate the minimum required number of nodes in Step 4.

3.3.1 Coordinates of the Nodes in a Pattern

We use equilateral triangle, square, and hexagon polygons for solving the target tracking problem. Initially, we divide the FoI into the regular polygons. We calculate the coordinates of the nodes where the nodes can place inside the FoI. Such coordinates help to estimate the number of required nodes in the next step. Fig. 3.4 illustrates a FoI that divides into l width and l_p height polygons, where l is the side length in a pattern p and $l_p = l\sin(\theta_p)$. The angle θ_p for equilateral triangle, square, and hexagon polygons are $\pi/3$, $\pi/2$, and $\pi/3$, respectively.

To estimate the coordinates of the nodes in a triangle pattern, the FoI is divided into even and odd number rows, denoted by r, as shown in Part (a_1) of Fig. 3.4. In an even row, *i.e.*, $r = (0, 2, 4, ...)$, the x and y coordinates of a sensor node are rl and rl_p, respectively. Similarly, we add $l/2$ in x coordinates of sensor nodes when r is odd. The x and y coordinates of the sensor nodes are given by

$$(x, y) \in \begin{cases} (rl, rl_p), & \textit{if } r \textit{ is even} \\ (rl + \frac{l}{2}, rl_p), & \textit{otherwise} \end{cases} \tag{3.2}$$

The sensor nodes in a square pattern have equal distance in x and y axis as shown in part (b_1) of Fig. 3.4. The x and y coordinates of the sensor nodes are given by

$$(x, y) \in (rl, rl_p) \tag{3.3}$$

The difference between triangle pattern and hexagon pattern is only that a sensor node is not located at the center of a hexagon pattern as shown in part (c_1) of Fig. 3.4. To estimate the coordinates of the sensor nodes in a hexagon pattern, we use Eq. 3.2 with the condition that the sensor nodes are not present in the center of the hexagon pattern, *i.e.*, **if** $mod(y/l_p, 2) = mod(x, 3)+1$ **then** not place a sensor node. From Eq. 3.2 and Eq. 3.3,

the coordinates of a sensor node in a pattern p (triangle, square, or hexagon) are given by

$$(x, y) \in \left(rl + l \cos\left(\frac{\theta_p}{2}\right) \cos(\theta_p) \, mod \left(\frac{y}{l_p}, 2\right), rl_p \right). \tag{3.4}$$

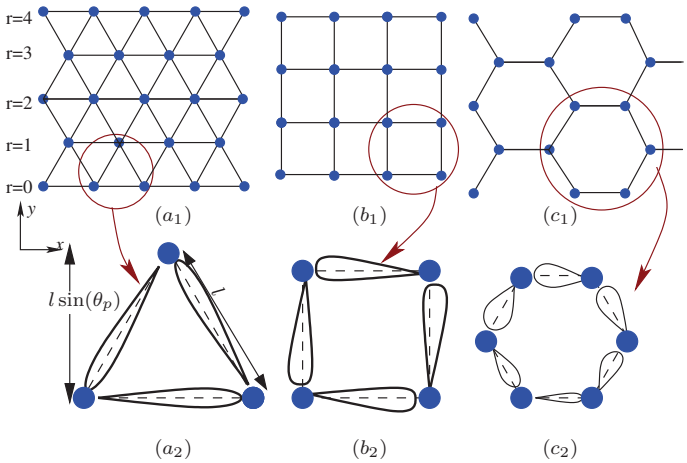

Figure 3.4 Regular sensor node placement patterns. Parts (a_1), (b_1), and (c_1) illustrate the triangle, square, and hexagon sensor node placement pattern, respectively. A filled circle (•) indicates the position of the sensor nodes in a given pattern.

3.3.2 Orientation of the Tracking Region

We use the following lemmas to determine the orientation of the tracking region of the nodes in a given regular pattern.

Lemma 3.1 *Let a FoI Ψ is divided into a square pattern of side length l. Let $\theta_{x,y}$ be the orientation of a sensor node located at (x, y), where $\{x, y\} \geq 0$. The FoI is k-target tracking if*

$$R_s \geq 2kl \ \textbf{\textit{and}} \ \theta_{x,y} = mod(x + y, 2) \times \frac{\pi}{2}. \tag{3.5}$$

Proof: *The lemma consists two conditions. The first condition shows the relation between the side length and the tracking range. The second condition illustrates the orientation of the tracking region of the sensor nodes.*

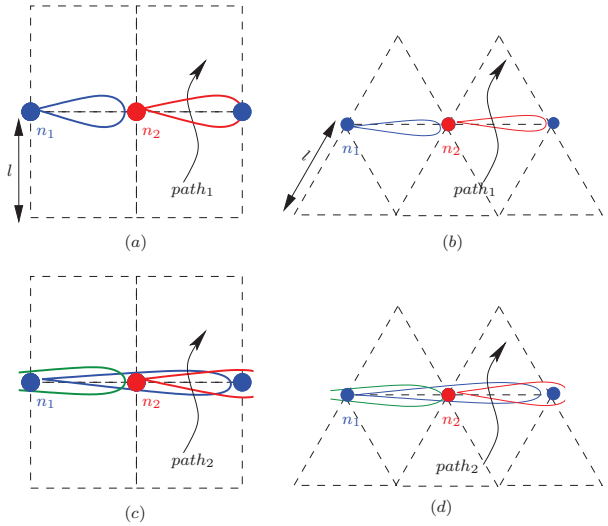

Figure 3.5 Illustration of Scenarios of 1-target tracking and 2-target tracking. Parts (a) and (b) show that a target is tracked by n_2 sensor node (1-target tracking of $path_1$) in square and triangle patterns, respectively. Parts (c) and (d) show that a target is tracked by n_1 and n_2 sensor nodes (2-target tracking of $path_2$) in square and triangle patterns, respectively.

• $\mathbf{R_s \geq 2kl}$*: Fig. 3.6 illustrates the locations and tracking regions of the sensor nodes in a FoI* Ψ*. Let* l *be the side length of a square polygon. Part (a) of Fig. 3.6 illustrates that the polygon has four sensor nodes and four tracking regions. Part (b) of the figure shows the Scenario when we add sensor nodes in* x *and* y *directions or increase the side length from* l *to* $2l$*. It shows that the number of sensor nodes and tracking regions are nine and* 12*, respectively. Similarly, part (c) illustrates the Scenario when the side length, number of sensor nodes, and tracking regions are* $3l$*,* 16 *and* $24l$*, respectively. In summary, the side length, number of sensor nodes, and tracking regions are given by*

$$\begin{bmatrix} \textit{Side length} \\ \textit{No. of sensor nodes} \\ \textit{Tracking regions} \end{bmatrix} = \begin{Bmatrix} l & 2l & 3l & 4l & \dots \\ 4 & 9 & 16 & 25 & \dots \\ 4l & 12l & 24l & 40l & \dots \end{Bmatrix} \tag{3.6}$$

The number sequence $\big(4, 12, 24, 40, 2j(j+1), \dots\big)$ *are known as four times triangular numbers, where an integer* $j = \{1, 2, 3, 4, \dots\}$*. Let the tracking region* $R_s = l \times f$ *and*

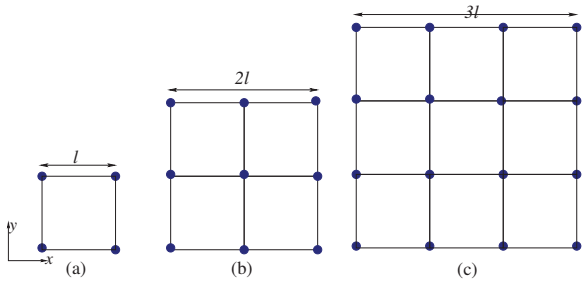

Figure 3.6 Tracking regions and sensor nodes in a square pattern. Parts (a), (b), and (c) illustrate l, $2l$, and $3l$ side lengths of square patterns, respectively.

integer $f > 0$. *Substituting* R_s *in Eq. 3.6, we get*

$$(j+1)^2 lf \geq 2j(j+1)l. \tag{3.7}$$

Eq. 3.7 is possible only when $f \geq 2$ *and* $j > 0$. *Since* $R_s = l \times f$ *and* $f \geq 2$ *and therefore* $R_s \geq 2l$.

Next, Definition 4 shows that a target in the FoI is tracked by k sensor nodes only if it lies within the tracking region of at least k sensor nodes. Parts (c) and (d) of Fig. 3.5 illustrate 2*-target tracking of* $path_2$ *in the FoI. It shows that the k-target tracking requires the k times the tracking range of* 1*-target tracking, i.e.,* $R_s \geq 2lk$.

• ***Orientation of tracking region****: The possible orientation of the tracking region in square pattern are* 0 *and* $\frac{\pi}{2}$. *In a square pattern, if we alternate the orientations of the sensor nodes then there is no overlap tracking region in the FoI and therefore the orientation of a sensor node at* (x, y) *is given by*

$$\theta_{x,y} = mod((x+y), 2) \times \frac{\pi}{2}. \tag{3.8}$$

□

Lemma 3.2 *Let a FoI* Ψ *is divided into a triangular pattern of side length* l. *Let* $\theta_{x,y}$ *be the orientation of a sensor node located at* (x, y). *The FoI is k-target tracking if*

$$R_s \geq 3kl \ \textbf{\textit{and}} \ \theta_{x,y} = mod((x + y + mod(y, 2)), 3) \times \frac{\pi}{3}. \tag{3.9}$$

Proof: *Similar to Lemma 3.1, we have the following two conditions:*

• $\mathbf{R_s \geq 3kl}$*: Fig. 3.4 illustrates the locations and tracking regions of the sensor nodes in a FoI* Ψ*. Let* l *be the side length of a triangle polygon. The minimum number of sensor nodes and tracking regions are* 3 *and* $3l$*, respectively. In summary, the side length, number of sensor nodes, and tracking regions are given by*

$$\begin{bmatrix} \textit{Side length} \\ \textit{No. of sensor nodes} \\ \textit{Tracking regions} \end{bmatrix} = \begin{cases} l & 4l & 7l & 10l & \dots \\ 3 & 15 & 36 & 66 & \dots \\ 3l & 30l & 84l & 165l & \dots \end{cases} \tag{3.10}$$

The number sequences $\left(3, 15, 36, 66, \frac{(j+1)(j+2)}{2}, \dots\right)$ *and* $\left(3, 30, 84, 165, \frac{3j(j+1)}{2}, \dots\right)$ *are known as triangular and centered nonagonal numbers, respectively, where integer* $j = \{1, 4, 7, 10, \dots\}$*. Since we assume that a directional sensor node has one tracking region of* R_s *tracking range, where* $R_s = l \times f$ *and an integer* $f > 0$*. Substituting* $R_s = l \times f$ *in Eq. 3.10, we get*

$$\frac{(j+1)(j+2)lf}{2} \geq \frac{3j(j+1)f}{2}. \tag{3.11}$$

Eq. 3.11 is possible only when $f \geq 3$ *and* $j > 0$*. Since* $R_s = l \times f$ *and* $f \geq 3$ *and therefore* $R_s \geq 3l$*. For* k*-target tracking, the tracking range* $R_s \geq 3kl$*, where* $k \geq 1$*.*

• ***Orientation of tracking region****: The possible orientation of the tracking region in triangle pattern are* $\{0, \frac{\pi}{3}, \frac{2\pi}{3}\}$*. From Fig. 3.4, the orientation of a sensor node at* (x, y) *is given by*

$$\begin{aligned} \theta_{x,y} &= \begin{cases} mod((x+y), 3) \times \frac{\pi}{3}, & \textbf{\textit{if}}\ \textit{row } r \textit{ is even} \\ mod((x+y+1), 3) \times \frac{\pi}{3}, & \textit{otherwise} \end{cases} \\ &= mod((x+y+mod(y,2)), 3) \times \frac{\pi}{3}. \end{aligned} \tag{3.12}$$

□

Lemma 3.3 *Let a FoI* Ψ *is divided into a hexagon pattern of side length* l*. Let* $\theta_{x,y}$ *be the orientation of a sensor node located at* (x, y)*, where* $(x, y) \geq 0$*. The FoI is* k*-target*

tracking if

$$R_s \geq 4kl \ \textbf{\textit{and}} \ \theta_{x,y} = mod((x + y + mod(y, 2)), 3) \times \frac{\pi}{3}. \tag{3.13}$$

Proof: *Similar to Lemma 3.1 and Lemma 3.2, we have the following two conditions:*

• $\mathbf{R_s \geq 4kl}$*: From Fig. 3.4, we have*

$$\begin{bmatrix} \textit{No. of sensor nodes} \\ \textit{Tracking regions} \end{bmatrix} = \begin{cases} 6 & 24 & 54 & 96 & \dots \\ 6l & 30l & 72l & 132l & \dots \end{cases} \tag{3.14}$$

The number sequence $\big(6, 30, 72, 132, 3j \times (3j-1), \dots\big)$ *is known as pentagonal numbers, where an integer* $j = \{1, 2, 3, 4, \dots\}$*. Let the tracking region of* R_s *tracking range is* $l \times f$*, where* $l > 0$ *and integer* $f > 0$*. Substituting* $R_s = l \times f$ *in Eq. 3.14, we get*

$$6j^2 \times l \times f \geq 3j \times (3j-1). \tag{3.15}$$

Eq. 3.15 is possible only when $f \geq 2$*. However, the hexagon pattern uses only half the tracking region of a sensor node. Therefore* $R_s \geq 4kl$*.*

• ***Orientation of tracking region****: The possible orientation of the tracking region in hexagon pattern is same as a triangle, i.e.,* $\{0, \frac{\pi}{3}, \frac{2\pi}{3}\}$*. However, the position of the sensor nodes in hexagon pattern is different from triangle pattern. From Fig. 3.4, the orientation of a sensor node at* (x, y) *is given by*

$$\begin{aligned} \theta_{x,y} &= \begin{cases} mod((x+y), 3) \times \frac{\pi}{3}, & \textit{if row } r \textit{ is even} \\ mod((x+y+1), 3) \times \frac{\pi}{3}, & \textit{otherwise} \end{cases} \\ &= mod((x + y + mod(y, 2)), 3) \times \frac{\pi}{3}. \end{aligned} \tag{3.16}$$

□

Theorem 3.1 *Let a FoI* Ψ *is divided into a pattern* p *of side length* l*. The FoI* Ψ *is*

k-target tracking if

$$R_s \geqslant \begin{cases} 3kl & \textit{if } p\textit{=0} \\ 2kl & \textit{if } p\textit{=1} \\ 4kl & \textit{if } p\textit{=2} \end{cases} \geqslant \left(\frac{3p^2 - 5p + 6}{2}\right) kl \tag{3.17}$$

and

$$\theta_{x,y} \geqslant \begin{cases} mod((x + y + mod(y, 2)), 3) \times \theta_0 & \textit{if } p\textit{=0} \\ mod((x + y), 2) \times \theta_1 & \textit{if } p\textit{=1} \\ mod((x + y + mod(y, 2)), 3) \times \theta_2 & \textit{if } p\textit{=2} \end{cases} \tag{3.18}$$

Proof: *Use the values of R_s and $\theta_{x,y}$ from Eqs. 3.5, 3.9, and 3.13 for square, triangle, and hexagon patterns, respectively, and hence proved.* □

3.3.3 Optimal Side Length in Regular Patterns

Next, we estimate the optimal side length in a given placement pattern for solving the target tracking problem. We assume that the location of a sensor node s in the FoI for a given pattern is denoted by $\mathbf{s}(x, y)$. We estimate the distance of m^{th} nearest neighbors of a sensor node that located at v, denoted by $v_{p,m}$. To estimate an optimal side length in a regular pattern that provides k-target tracking and m-connectivity in a given segment length, the solution should satisfy the following conditions simultaneously:

- **k-target tracking:** Theorem 3.1 gives the relationship between the tracking range and a pattern p, *i.e.*, $R_s \geq \left(\frac{3p^2-5p+3}{2}\right) kl$.
- **m-connectivity:** The relation between the communication range and the m^{th} communication neighbor of a sensor node s is $v_{p,m}$.

- **Segment length:** Eq. 3.1 shows the relationship between the segment length in a pattern p, *i.e.*, *segment length* $\geq l\sqrt{p+1}$.

The optimal side length for solving the target tracking problem in a regular pattern p, denoted by O_p, is therefore

$$O_p = \min\left(\frac{\text{segment length}}{\sqrt{p+1}}, \frac{2R_s}{(3p^2-5p+6)k}, \frac{R_c}{v_{p,m}}\right). \tag{3.19}$$

3.3.4 Placement of the Sensor Nodes in the FoI

In this section, we use the optimal side length in a polygon for placement of the sensor nodes in the FoI. We consider a practical Scenario where the shape of the FoI is not a regular. An example Scenario of an irregular-shaped FoI Ψ is illustrated in Fig. 3.7. For placement of the sensor nodes, we convert irregular-shaped FoI into the regular-shaped FoI. Fig. 3.7 illustrates that a L width and B length rectangular-shaped regular region is created around an irregular-shaped FoI. Parts (a1) and (b1) of Fig. 3.7 illustrate the placement of the sensor nodes in triangle pattern and square pattern, respectively. Parts (a2) and (b2) illustrate the placement of sensor nodes after displacement along the boundary of the FoI. Let N_p denotes the required number of sensor nodes for solving target tracking problem in a given pattern p. From Fig. 3.4, we have

$$\begin{aligned}
N_0 &= \frac{L}{O_0 \times \sin(\theta_0)}\left(\frac{B}{O_0} - \frac{1}{2}\right),\\
N_1 &= \frac{L}{O_1} \times \frac{B}{O_1},\\
N_2 &= \frac{L}{O_2 \times \sin(\theta_2)}\left(\frac{B}{O_2} - \frac{1}{2} - \frac{B}{3 \times O_2}\right).
\end{aligned}$$

In summary, Eq. 3.20 evaluates the required number of sensor nodes, *i.e.*,

$$N_p = \left(\frac{L}{O_p \sin(\theta_p)}\right)\left(\frac{B}{O_p} - \frac{Bp\cos(\theta_p)}{3O_p} - \cos(\theta_p)\right). \tag{3.20}$$

Algorithm 1: Location of sensor nodes, optimal pattern, and optimal side length for k-target tracking in m-connected WSNs.

Input: Tracking range, communication range, m, k, segment length, and FoI;

Output: Location of sensor nodes, optimal pattern, optimal side length;

Set $d \leftarrow 1$, $Result_p \leftarrow 5000$;

for $p \leftarrow 0$ **to** 2 **do**

Estimate side length of a regular pattern p for a given segment length using Eq. 3.1, *i.e.*, $l \leq \frac{\text{Segment length}}{\sqrt{p+1}}$.

Estimate side length of a regular pattern p for a given segment length using Theorem 3.1, *i.e.*, $l \leq \frac{2R_s}{(3p^2-5p+6)k}$.

Estimate the optimal side length in a given pattern p using Eq. 3.19, *i.e.*,

$$O_p = \min\left(\frac{\text{segment length}}{\sqrt{p+1}}, \frac{2R_s}{(3p^2-5p+6)k}, \frac{R_c}{v_{p,m}}\right).$$

Estimate the required number of sensor nodes for a given pattern p using Eq. 3.20, *i.e.*,

$$N_p = \left(\frac{L}{O_p \sin(\theta_p)}\right)\left(\frac{B}{O_p} - \frac{Bp\cos(\theta_p)}{3O_p} - \cos(\theta_p)\right).$$

if $N_p \leq Result_p$ **then**

$Optimal_{pattern} \leftarrow p$;

$Optimal_{distance} \leftarrow O_p$;

$Result_p \leftarrow N_p$;

Set $p = Optimal_{pattern}$, $l = Optimal_{distance}$, and $l_p = l\sin(\theta_p)$. **for** $x \leftarrow 0$ ***to*** L *and* $y \leftarrow 0$ ***to*** B **do**

Estimation position of a sensor node at $s(x, y)$ using Eq. 3.4, *i.e.*,

$$s(x,y) \in \left(rl + l\cos\left(\frac{\theta_p}{2}\right)\cos(\theta_p)\, mod\left(\frac{y}{l_p}, 2\right), rl_p\right),$$

Estimate tracking direction of a sensor node in a given pattern p using Theorem 3.1, *i.e.*,

$$\theta_{x,y} \geqslant \begin{cases} mod((x+y+mod(y,2)),3)\times\theta_0 & \text{if p=0} \\ mod((x+y),2)\times\theta_1 & \text{if p=1} \\ mod((x+y+mod(y,2)),3)\times\theta_2 & \text{if p=2} \end{cases}$$

if $(x, y) \in \Psi$ **then**

Place the sensor node at $s(x, y)$ with direction $\theta_{x,y}$.

if $(x, y) \notin \Psi$ **then**

Place the sensor node on the intersection location of the boundary of the FoI and the direction $\theta_{x,y}$.

if $\big((mod(y,2)=mod(x,3)+1)$ *and* $p=2\big)$ **then**

$x \leftarrow x + l$;

$x \leftarrow x + l$;

$y = y + l_p$;

Return Location of sensor nodes, $Optimal_{pattern}$, and $Optimal_{distance}$;

The complete procedure for solving the target tracking problem is shown in Algorithm 1. Definition 4 shows that a minimum directional sensor nodes required pattern is known as optimal pattern in the TTS. We use the algorithm for selecting an optimal pattern for solving TTS.

3.3.5 Practical Challenges

The deployment of sensor nodes for solving the tracking problem considers some practical challenges, such as the estimated location of a sensor node is outside the FoI but the tracking region of the sensor node is partially required as shown in part (a) of Fig. 3.7. In such Scenarios, the proposed solution first calculates the location and tracking direction of the sensor node. Next, the solution locates the sensor node on the intersection location of the boundary of the FoI. Parts (a2) and (b2) of Fig. 3.7 illustrate such Scenarios where sensor nodes are located on the boundary of the FoI. The other practical challenge is the presence of the obstacles in the FoI. The proposed work is suitable when there is no obstacle or partition in the FoI. The proposed work can be used when the obstacles are static or pre-fixed in the FoI. In such Scenarios, we can divide the FoI into regions and use the Algorithm 1 for each subregion to solve the tracking problem.

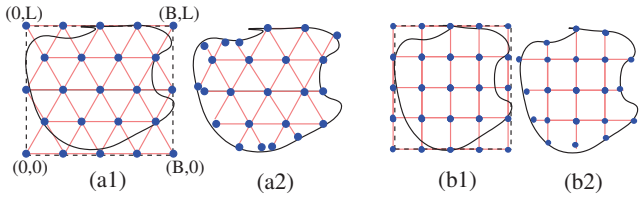

Figure 3.7 Regular patterns with the FoI. Solid and dotted lines illustrate the FoI and rectangular-shaped boundary of the FoI, respectively.

Example: Let the values of tracking range R_s, communication range R_c, segment length, k, m, L, B, and the area of the rectangular-shaped FoI Ψ be $35m$, $50m$, $8m$, 2, 1, $80m$, $50m$, and 4×10^3 m^2, respectively. We use Eq. 3.19 and Eq. 3.20 for estimating the optimal side length and the number of sensor nodes, respectively. We get $p = 1$, $O_1 = 5.65m$, and

$N_1 = 125$. It shows that a square-shaped polygon of side length *5.65m* provides 2-target tracking and a connected network. The coordinates of the sensor nodes are *(0,0)*, *(0,5.65)*, . . . , *(0,74.35)*,. . . ,*(44.35,74.35)*.

3.4 Simulation and Numerical Results

In this section, we introduce the network and target mobility models. We analyze the performance of our algorithm using simulation and numerical results. We used Network Simulator 2.34 (NS 2.34) for discussing the simulation results.

• **Network model:** Wireless Scenarios are not directly supported in NS 2.34 simulator. Therefore, we used the CMU wireless extension CMU (2014). We modified the radio propagation model (wirelessphy.cc file in NS2.34) for creating the application layer. As shown in Section 3.2, the sensor nodes in WSNs have various parameters such as tracking range, tracking direction, tracking angle, etc. Parameters of tracking model are shown in Table 3.2.

• **Target mobility model:** In this work, we use the following motility models:

(a) *Random walk mobility model:* In this model, a target moves from its current location to a new location by randomly choosing the direction and speed. The new speed and direction are chosen from pre-defined ranges. Such ranges of the speed and direction are denoted by $[\text{speed}_{min}, \text{speed}_{max}]$ and $[0, 2\pi]$, respectively. The average speed of a target in this model is denoted by speed_{avg}, where $\text{speed}_{min} \leq \text{speed}_{avg} \leq \text{speed}_{max}$. Each movement occurs in either a constant time interval or with a constant distance traveled and then with a different direction and/or speed. Let the constant time interval and distance traveled are denoted by C_t and C_d, respectively.

(b) *Random waypoint mobility model:* This model is characterized by the pause time of the moving target, denoted by P_t. A target stays in the same location for P_t and subsequently, chooses a random destination in the FoI and moves with a speed uniformly distributed between [speed_{min} and speed_{max}]. The average speed of a target in this model

is denoted by speed_{avg}, where $\text{speed}_{min} \leq \text{speed}_{avg} \leq \text{speed}_{max}$.

Table 3.2 Network Parameters.

Parameter	**Value**
Tracking Angle	30°, 45°, 60°
Tracking Range	$5m$, $7m$, $10m$
Communication Range	$10m$, $15m$
Uniform Error	1.0 for 60°, 0.75 for 45°, 0.05 for 30°
Speed_{avg} (in m/sec)	$1.0, 2.0, 3.0$
Random Waypoint P_t	120 seconds
Random walk C_t	180 seconds
Random walk C_d	130 meters
Area of FoI $\|\Psi\|$ (in m^2)	$200^2, 220^2, \ldots, 300^2$
No. of moving target in FoI	1

3.4.1 Impact of Tracking Angle

In this result, we illustrate the impact of the tracking angle of directional sensor nodes on the target tracking error. The target tracking error is defined as the difference between exact location and tracked location of the target. For estimating the average target tracking error, we randomly picked 100 tracking points in the FoI and take the average of the target tracking error. Table 3.3 illustrates the impact of tracking angle on average tracking error in meter. An interesting observation from this result is that the tracking error increased with tracking angle and speed of the target. This is because, the stability of a target reduces with increases the speed. Another interesting observation is that the triangle pattern provides minimum average target tracking error. This is because the triangle pattern consists of the minimum distance between two sensor nodes in the pattern. Eq. 3.1 also illustrates the same thing. However, the minimum distance between sensor nodes in triangle pattern increases the density of sensor nodes (sensor nodes per unit area) in the FoI and therefore increase the cost of the network.

Table 3.3 Impact of tracking angle on average tracking error (in meter).

	Random walk			**Random waypoint**		
	Target speed (m/s)			**Target speed (m/s)**		
	1	**2**	**3**	**1**	**2**	**3**
30°	0.405	0.414	0.432	0.432	0.45	0.468
45°	0.477	0.504	0.531	0.504	0.54	0.567
60°	0.549	0.567	0.594	0.576	0.603	0.63

(a) Triangle placement pattern.

	Random walk			**Random waypoint**		
	Target speed (m/s)			**Target speed (m/s)**		
	1	**2**	**3**	**1**	**2**	**3**
30°	0.451	0.462	0.487	0.481	0.502	0.527
45°	0.531	0.564	0.591	0.561	0.604	0.631
60°	0.613	0.636	0.662	0.643	0.676	0.702

(b) Square placement pattern.

	Random walk			**Random waypoint**		
	Target speed (m/s)			**Target speed (m/s)**		
	1	**2**	**3**	**1**	**2**	**3**
30°	0.495	0.506	0.528	0.528	0.55	0.572
45°	0.583	0.616	0.649	0.616	0.66	0.693
60°	0.671	0.693	0.726	0.704	0.737	0.77

(c) Hexagon placement pattern.

3.4.2 Impact of Tracking Range

Next experiment illustrates the impact of tracking range on average tracking error in meters. The tracking range of the sensor node is increased from $5m$ to $10m$ in this experiment. Random walk and waypoint mobility models with 45° tracking angle was considered to observe the effect of tracking range on target tracking error. Table 3.4 shows that the average tracking error increased by 50% when the tracking range is increased from $5m$ to $10m$. This is because the larger tracking range reduces the tracking signal strength. The larger tracking range also increases the overlapping tracking regions and sensor node interference.

Table 3.4 Impact of tracking range on average tracking error (in meter).

	Random walk			**Random waypoint**		
	Target speed (m/s)			**Target speed (m/s)**		
	1	**2**	**3**	**1**	**2**	**3**
$5m$	0.37	0.37	0.39	0.39	0.41	0.43
$7m$	0.43	0.46	0.48	0.45	0.49	0.51
$10m$	0.50	0.52	0.54	0.52	0.55	0.57

(a) Triangle pattern.

	Random walk			**Random waypoint**		
	Target speed (m/s)			**Target speed (m/s)**		
	1	**2**	**3**	**1**	**2**	**3**
$5m$	0.41	0.42	0.44	0.43	0.45	0.47
$7m$	0.48	0.51	0.53	0.50	0.54	0.57
$10m$	0.55	0.57	0.60	0.58	0.61	0.63

(b) Square pattern.

	Random walk			**Random waypoint**		
	Target speed (m/s)			**Target speed (m/s)**		
	1	**2**	**3**	**1**	**2**	**3**
$5m$	0.45	0.46	0.48	0.48	0.50	0.52
$7m$	0.53	0.56	0.59	0.56	0.60	0.62
$10m$	0.61	0.63	0.66	0.64	0.67	0.69

(c) Hexagon pattern.

3.4.3 Impact of k-Target Tracking

We study the impact of k on the number of required sensor nodes for k-target tracking in connected WSNs. Fig. 3.8 shows an increase in the required number of sensor nodes for 1-connectivity, R_c=45m, R_s=35m, $\|\Psi\|$=200m×200m, segment length=8m, and $1 \leq k \leq 4$. The numerical results are represented by solid lines and the simulation results are using 97% confidence interval. Obviously, as the value of k increases, the required number of sensor nodes increases. It can be seen in Fig. 3.8 that the expected k-target tracking calculated with simulation, matches closely with that evaluated results of Algorithm 1. This validates the proposed work. The result shows that the square pattern requires a minimum number of sensor nodes in the network except $k = 1$. This is because, the higher value of k increases more overlapping of tracking regions and therefore requires more

number of sensor nodes. The triangle, square, and hexagon patterns have three, four, and three neighbors, respectively, as shown in Fig. 3.4. The more number of neighbors utilize the overlapping of tracking regions and as a result reduces the required number of sensor nodes.

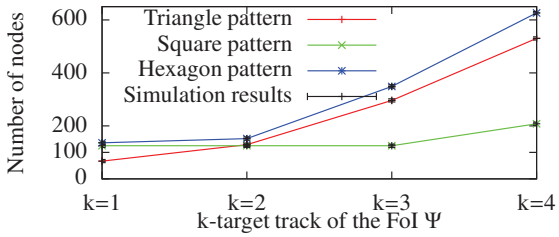

Figure 3.8 Impact of k-target tracking on number of sensor nodes, where k={1,2,3,4}.

Next, we show the impact of k on the tracking range of a sensor node. Fig. 3.9 illustrates the impact of k on the tracking range of the sensor nodes for 1-connectivity, R_c=45m, $\|\Psi\|$=200$m \times$ 200m, segment length=8, and $1 \leq k \leq 4$. We assumed that 200 sensor nodes are placed in the FoI for k-target tracking. Fig. 3.9 shows the simulation results with 98% confidence interval. The result shows that the tracking range of a sensor node is very high for higher value of k. This is because the higher value of k requires larger tracking region in a fixed number of sensor nodes and tracking angle. It also shows that square deployment pattern requires higher tracking range than hexagon pattern and therefore consumes more energy.

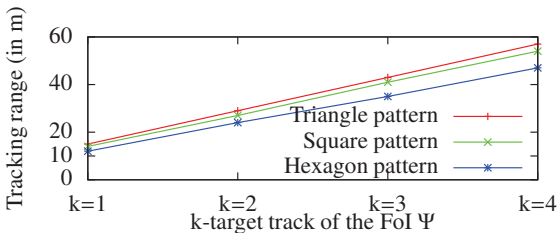

Figure 3.9 Relationship between k-target tracking and tracking range.

3.4.4 Compare With Existing Work

Finally, we compare the proposed work with an exiting work. Derr and Manic Derr and Manic (2015) considered the triangle deployment pattern of the sensor nodes in a given FoI for 1-target tracking. Fig. 3.10 shows the simulation results for area of the FoI=$\{3000, 3200, 3400, 3600, 3800, 4000\}$ m^2. In this simulation result, we considered k=1, k=2, R_c=45m, R_s=35m, and segment length=18m. It illustrates that the number of sensor nodes increases with the area of the FoI when $k = 1$ and $k = 2$. The result also illustrates that square pattern requires the least number of sensor nodes when k=1 and k=2. The number of neighbors of a sensor node in square, triangle, and hexagon patterns are four, three, and three, respectively. The square has maximum number of neighbors and therefore provides more contributions for target tracking. The result shows that the cost of the proposed TTS is less as compare to the triangle deployment pattern based TTS Derr and Manic (2015). It can be seen in Fig. 3.10 that the number of sensor nodes estimated using simulation results match with Algorithm 1.

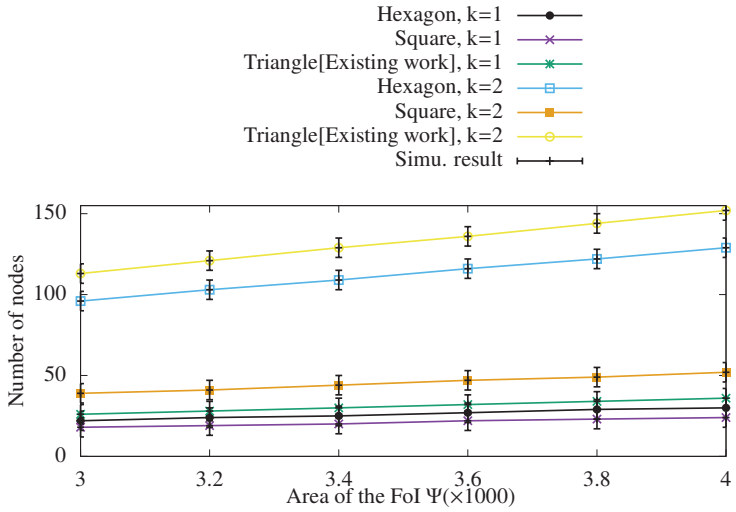

Figure 3.10 Impact of area of the FoI on number of directional sensor nodes for different values of k-target tracking.

Fig. 3.11 shows an increase in the number of sensor nodes with tracking range. An interesting observation from Fig. 3.11 is that square pattern requires least number of

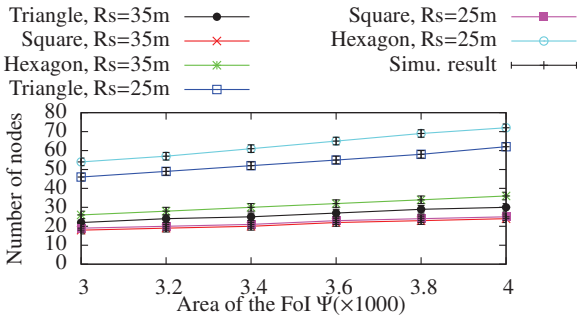

Figure 3.11 Impact of area of the FoI on number of directional sensor nodes for different values of tracking range.

sensor nodes for $R_s = 25m$ and $R_s = 35m$. Similar as previous result, this is because a sensor node in the square pattern has four neighbors while the triangle and hexagon patterns consist only three neighbors.

3.5 Target Tracking System

In this section, we developed a prototype to demonstrate how the proposed analysis solves the target tracking problem. We developed a Target Tracking System (TTS) for tracking the moving targets in the FoI.

3.5.1 Overview of TTS

TTS uses Raspberry PI 3 board with camera nodes for tracking a target and communicating the tracking information to the base station. The Raspberry PI 3 is the third generation Raspberry PI board. It has replaced the Raspberry Pi 2 Model B in February 2016. It is equipped with a 1.2GHz 64-bit quad-core ARMv8 CPU, 802.11n Wireless LAN, Bluetooth 4.1, Bluetooth Low Energy (BLE), camera interface, display interface, Micro SD card slot, and VideoCore IV 3D graphics core. The Raspberry PI camera module is capable of taking full HD 1080p photo and video and can be controlled programmatically. We used a server system as a data collection sensor node, called base station in TTS. The server received the tracking information (images of targets) from Raspberry PI using BLE

and transmitted the information to the client using WiFi. TTS uses Android operating system based smartphone for display the target information to the client. Figs. 3.13, 3.14, and 3.15 illustrate the configuration of TTS for three different Scenarios with the tracking camera, data collection sensor node, and base station. The TTS used BLE and Wi-Fi protocols to form a connected WSN.

3.5.2 Working Principle of TTS

The TTS used the camera as directional sensor node for detecting the target in the FoI. A camera has fixed tracking angle and tracking range. Fig. 3.12 illustrates the deployment of the sensor nodes in the FoI. In a deployment Scenario of TTS, sensor nodes (Raspberry PI and camera) are located in the FoI, where the location of each sensor node and the direction of tracking region (camera focus) is calculated using Algorithm 1. Once a camera captures an image of a target, it is transfered to the Raspberry PI sensor node. A Raspberry PI uses BLE to forward the tracking information to the server. The server communicates the useful information of the target to the user using WiFi. Figs. 3.13, 3.14, and 3.15 illustrate three Scenarios of TTS.

Figure 3.12 Experiment setup at IIT (BHU) Varanasi. Leftside and right-side of the figure illustrate the placement of sensor nodes in a computer lab and the corridor of the department of computer Science & Engineering, respectively. The sensor nodes are connected with server and server communicates with the user using BLE and WiFi, respectively.

3.5.3 Placement Scenarios of TTS

In this work, we designed the following three Scenarios for placement of sensor nodes to solve the target tracking problem.

• **Indoor environment with no placement constraints (Scenario 1):** In this Scenario, we deployed TTS in a computer lab, a photograph of which is shown in Fig. 3.12. We assume that there is no obstacle in the FoI. Fig. 3.13 illustrates the Scenario. We used Algorithm 1 and estimated the distance between two sensor nodes, direction of tracking region, and the location of the sensor nodes in the FoI. The area of the FoI, communication range, tracking angle, k, m, and tracking range are $50m^2$, $50m$, $45°$, 1, 1, and $6m$, respectively. Substituting these values in Algorithm 1, we get the number of sensor nodes required for target tracking is 20 and deployment pattern is square. Hence, we placed 20 sensor nodes in the square pattern for tracking a target and transfer the tracking information to the client.

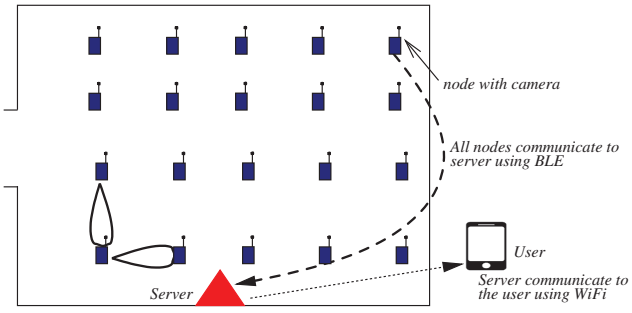

Figure 3.13 Illustration of Scenario 1. Placement of sensor nodes in Computer lab. The sensor nodes are connected with server and server communicates with the user using BLE and WiFi, respectively.

• **Indoor environment with placement constraints (Scenario 2):** In this Scenario, we deployed TTS in the corridor of the department to track incoming and outgoing students, a photograph of which is shown in Fig. 3.14. Different from the previous Scenario, the corridor region has obstacles and the shape is also not suitable for direct placement of the directional sensor nodes. Therefore, we have some placement constraints in such

Scenarios. In this Scenario, we first use Algorithm 1 to estimate the position of the sensor nodes in the corridor. We then rearrange the sensor nodes in possible location in the FoI. The estimated position of the sensor nodes and the rearrange position of the sensor nodes are shown by red and green color circles in Fig 3.14, respectively.

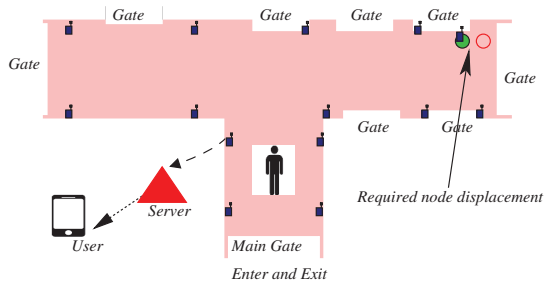

Figure 3.14 Illustration of Scenario 2. Placement of sensor nodes in corridor.

• **Outdoor Environment (Scenario 3):** In this Scenario, we deployed TTS on the road for tracking and counting the vehicles. Fig. 3.15 illustrates the deployment Scenario where sensor nodes are deployed along the road and communicated the information to the client.

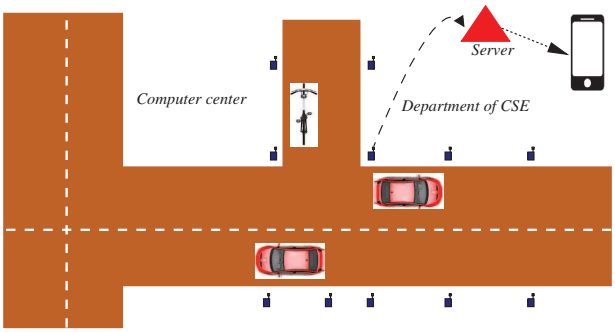

Figure 3.15 Illustration of Scenario 3. Placement of sensor nodes along the road.

3.5.4 Experimental Results

In this section, we show the results of the field experiments. In the field experiments, several tests are conducted to determine if a target located a point in the FoI selected

at random is tracked by at least one directional sensor node. We considered all three Scenarios as shown in Figs. 3.13, 3.14, 3.15. A test is termed successful if the target is tracked, otherwise it is termed unsuccessful. The number of such tests conducted is equal to an integer value closer to its area and repeat the experiment 100 times by changing the points. The tracking probability of a target in a given region is computed as the ratio of the number of tests successful to the number of tests conducted. Table 3.5 shows the experiment results in the form of average target tracking probability. It can be observed from Table 3.5 that the square deployment pattern provides highest tracking probability in all Scenarios. This is because, it requires lease number of sensor nodes in the FoI.

Table 3.5 Average Target Tracking probability.

Scenarios	**Segment length**	**Placement Patterns**		
		Triangle	**Square**	**Hexagon**
Scenario 1	**5m**	98.9	99.1	97.9
	7m	98.2	98.8	97.5
	9m	98.0	98.4	97.1
Scenario 2	**5m**	97.4	98.2	97.0
	7m	97.1	97.8	96.7
	9m	96.9	97.2	96.2
Scenario 3	**5m**	98.2	98.7	97.3
	7m	97.8	98.1	96.7
	9m	97.2	97.6	96.1

Next, Scenario 2 deployed TTS in the corridor of the department. However, this Scenario has some placement constraints and therefore we rearranged the sensor nodes in possible location for reducing the target tracking error. Table 3.6 shows the results in form of average required displacement of sensor nodes in Scenario 2. It illustrates that square placement pattern required least displacement compare with triangle and hexagon patterns.

Finally, TTS deployed on road for counting the vehicles as shown in Fig 3.15. Table 3.7 compares the number of vehicles counted by TTS and the actual vehicles pass in each of the 10 experiments. It can be observed that the overall accuracy with which the vehicles were detected is above 98%. Table 3.7 shows that not all the vehicles were detected by

Table 3.6 Average required displacement.

	Segment length	**Placement Patterns**		
		Triangle	**Square**	**Hexagon**
Scenario 2	5m	0.8m	0.5m	1.0m
	9m	1.0m	0.6m	1.2m

TTS. This is because the vehicles take turns (U-turn). Also, the lighting problem in evening time decreases the accuracy of detection.

Table 3.7 Results of vehicles counting experiments.

Experiment No.:	1	2	3	4	5	6	7	8	9	10	**Total**
Counted by camera:	25	11	9	16	14	9	12	22	11	10	**139**
Counted by TTS:	24	10	9	14	13	11	11	24	9	10	**135**
Accuracy	**135/139=0.98**										

3.6 Conclusion

In this chapter, we solved the k-target tracking problem using directional sensor nodes in m-connected WSNs. We used triangle, square, and hexagon regular patterns for deployment of sensor nodes in the FoI. We determined the orientation of tracking region, a pattern which required minimum number of directional sensor nodes, optimal side length of the pattern, and location of the sensor nodes for solving target tracking problem. The correctness of the proposed analysis validated by numerical and simulation results. We demonstrated an application of the analysis to design a Target Tracking System (TTS). The TTS was deployed in the computer lab, corridor of the department, and along the road-side to gather tracking information. It does not involve excavation of the surface and reduces the effort in planning for WSNs. We believe that our analysis can be used in the planning and design of large scale WSNs with deterministically deployment of directional sensor nodes for target tracking in the FoI.