CHAPTER II

Preliminaries

In this chapter, we provide some background on mobility models, growth of functions, relevant complexity theory and graph theory. All of these items will be used in the later chapters.

2.1 Preliminaries

- Let $N = \{1,2,3,\ldots\}$.
- Let S be a set. The set S^k, for $k \in N$, is the k-fold Cartesian product of the set S.

2.2 Mobility in Wireless Networks

Wireless network can be divided into Cellular Networks, Mobile Ad Hoc Networks, and Sensor Networks as follows [17]:

2.2.1 Cellular Networks

Cellular networks are the static wireless sources which separate the communication area into cells. We usually concern about the protocols that keep tracking the movement of the nodes between cells without interrupting the communication, or use cell information to track the location of clients.

2.2.2 Mobile Ad Hoc Networks

A Mobile Ad Hoc Wireless Network (MANET) is a self-configuring network of mobile nodes, and the position of each node is often unavailable. We usually concern about the routing direction between sources and the target data while the intermediate are moving and battery is limited, or the routing message of a MANET to keep the routing table.

2.2.3 Sensor Networks

A sensor network is the network of sensor devices. A sensor network is used to detected data, temperature, pressure motion or sound, and propose how to send it to some operation stations.

2.3 Realistic Mobility Patterns

This is a real movement that are tracked from the movement of objects in reality. We can classify the mobility patterns of the realistic movement as follows:

2.3.1 Pedestrians

It is the walking patterns of people or animals which is slow and limited the speed. For example, people with mobile, pets with sensor attached, animal herd with limited energy sensors. It is mainly worked in a two-dimensional area and sometimes work with the group mobility,

2.3.2 Marine and Submarine Mobility

There are both two-dimensional and three-dimensional movements like the movement of sea surface and submarine, respectively. The speed is limited by the machine of the ships or the aquanaut movement. Water absorb high-frequency signal is one of the restrictions of the marine communication.

2.3.3 Earth Bound Vehicles

The wheels base movement. Most works are on only one dimension, along a street, a pathway or even a train rail. GPS sometimes used to track the vehicles. The landscape building, road sign reflection, and tunnels may decrease the signal quality. Some extreme group mobility patterns, like passenger travelling by train, may also cause the communication problem.

2.3.4 Aerial Mobility

The flying objects usually travel a long distance, flock of birds, and sometimes move at a high speed, airplane. Airplane generally preserves their height. The anti-collision, message passing, position tracking, and flight control applications are proposed.

2.3.5 Medium Based Mobility

An ad hoc use sensor network to get information about studying the medium, measure the mediums or currents. It can be worked in any dimension depend on the medium itself and the circumstances (gas, fluid, air). Mostly study on individual movements.

2.3.6 Mobility in Outer Space

Ad hoc will be used for these spaces movement. The movement patterns are complex, and there are many restrictions such as the planet's gravity, the orbit patterns, the spaceship acceleration, or the fuel. The networks are used for preventing the collision or for recalibrating the orbit positions.

2.3.7 Robot Motion

Robots can be moved in any pattern and dimension that the robot designer design. Most applications are the project to coordinate the robots tasks by using the communication via a network.

2.4 Synthetic Mobility Model

Synthetic mobility model is the model that use a mathematical equation to generalize a movement pattern. We classify the mobility model as follows:

2.4.1 Cellular Mobility Model

Most of the work in the cellular model involve the movement of the nodes between cells.

2.4.1.1 ***Random Walk Model***: A node move from cell to cell by some probability. It is a memoryless model which is used most in the researchers.

2.4.1.2 ***Trace based Models***: This model used a real data collected from some companies, and it is valuable for evaluating protocols by using a real situation and data. However, the trace data sometimes undisclosed.

2.4.1.3 ***Fluid Flow Mobility Model***: It represents the traffic flow by comparing the traffic movement with the flow of fluid in a pipe.

2.4.2 Random Trip Mobility

Some variant of MANETs models are shown as follows:

2.4.2.1 ***Random Walk Mobility Model***: It is the memoryless mobility model. A node chooses a random direction and speed at a time, with some bound.

2.4.2.2 ***Random Waypoint Mobility Model***: It is the random walk model which have stop time before a node change its direction or speed.

2.4.2.3 ***Random Direction Mobility Model***: The random waypoint mobility model that the nodes stop and randomly choose a new direction and velocity and travel to the direction in the working area.

2.4.2.4 ***A Boundless Simulation Area Mobility Model***: A model of torus space.

2.4.2.5 ***Gauss-Markov Mobility Model***: The tuning degree of this model is randomly varies according to the mobility pattern.

2.4.2.6 ***A Probabilistic Version of the Random Walk Mobility Model***: Next step movement is related to the previous steps movement with some probabilities.

2.4.2.7 ***City Section Mobility Model***: The random waypoint restricted by the street on a city map.

2.4.3 Group-Mobility Models

The mobility model that describes the group behavior. They are classified as follows:

2.4.3.1 ***Exponential Correlated Random Mobility Model***: The group mobility is set from a motion function.

2.4.3.2 ***Column Mobility Model***: The group mobility direction is set by the member in the group.

2.4.3.3 ***Nomadic Community Mobility Model***: The movement direction is nomadic set to let all nodes in the group move together from one place to another.

2.4.3.4 ***Pursue Mobility Model***: There is a target node, and the group member will follow the node.

2.4.3.5 ***Reference Point Group Mobility Model***: The group member follows a logical center that has its individual movement.

2.4.4 Particle Based Mobility

The model for predicting pedestrian behavior in a panic situation that force pedestrian movement with a group into some direction.

2.4.5 Combined Mobility Models

Many mobility models are collected to be considered.

2.4.6 Non-recurrent Models

Fully or partially predictable movement function defined to some mobile objects. It counts the changes in the geometric structure to maintain the movement function.

2.4.7 Worst-Case Mobility Models

Any movement of the nodes is limited by a velocity, acceleration, or a constant time.

2.4.8 Mobility Models and Reality

There are some conflicts among the real situation mobility and the theoretical one. The theoretical model usually assume working in an ideal situation without mistakes. Even, there are adaptations of a theoretical mobility model to some real situation; it is unproven that the model can describe the real situation. For example, the random trip mobility model is adapt to simulate the movement on a street map.

2.5 Definitions of Greenlaw and Kantabutra Mobility Model

The following definition of the mobility model is taken from [14] with permission from the authors, as is part of the ensuing discussion. We define the model here to operate on a two-dimensional grid. A mobility model is an 8-tuples (S, D, U, L, R, V, C, O), where

1. The set $S = \{s_1, s_2, ..., s_m\}$ is a finite collection of **sources**, where $m \in \mathrm{N}$. The value m is the **number of sources**. Corresponding to each source s_i, for $1 \le i \le m$, an **initial location** (x_i, y_i) is specified, where $x_i, y_i \in \mathrm{N}$.

2. The set $D = \{000, 001, 010, 101, 110\}$ is called the **directions**, and these values correspond to no movement, east, west, south, and north, respectively.

3. The set $U = \{u_1, u_2, ..., u_p\}$ is a finite collection of *mobile devices*, where $p \in \mathrm{N}$. The set U is called the set of **users**. The value p is called the **number of users**. Corresponding to each user u_i, for $1 \le i \le p$, an initial location (x_i, y_i) is specified, where $x_i, y_i \in \mathrm{N}$.

4. The set $L = \{l_1, l_2, ..., l_t\}$ is a finite collection of "bit strings," where $t \in \mathrm{N}$ and $l_i \in D^t$ for $1 \le i \le t$. The set L is called the set of **user movement**. Each group of three bits in l_i beginning with the first three defines a step in a given direction for the user u_i's movement or no movement at all if the string is 000. The value t is called the **duration of the model**.

5. The set $R = \{r_1, r_2, ..., r_m\}$ is a finite collection of "bit strings," where $r_i \in D^{t(i)}$, $t(i) \in \mathrm{N}$ and $1 \le i \le m$. Each group of three bits in r_i beginning with the first three defines a step in a given direction for the source s_i's movement or no movement at all if the string is 000. The set R is called the **random walks** or **source movement** of the mobility model.

6. The set $V = \{v_1, v_2, ..., v_m\}$ is a finite collection of numbers, where $v_i \in \mathrm{N}$. The value v_i is the corresponding number of steps from r_i per unit time that s_i will take. This set is called the **velocities**.

7. The set $C = \{c_1, c_2, ..., c_m\}$ is a finite collection of lengths, where $c_i \in \mathrm{N}$. The value c_i is the corresponding diameter of the circular coverage of source s_i. This set is called the **coverages**.

8. The set $O = \{(x_1, y_1, x_2, y_2) \mid x_1, y_1, x_2, y_2 \in \mathrm{N}, x_2 > x_1 \text{ and } y_2 > y_1\}$ is a finite collection of rectangles in the plane. This set is called the **obstacles**.

Greenlaw and Kantabutra design the theoretical mobility model in a two-dimensional grid for simplicity. However, the model can be extended to a three-dimensional grid, a finer grid, or some other shape of a grid for more reality. In the model, the sources in S are the wireless access points. They are always broadcasting and receiving signals. Although real world sources do not move in discrete steps, finer grid can make it realer. Greenlaw and Kantabutra assume that the sources are always at a grid point locations. The set D is the four possible directions that a source or user can move in the grid, include no movement. The set U is the set of users carry a mobile device. Users' movement are random and are specified by the series of directions contained in the set L. All users' velocity are the same, equal to one. All user will travel by a unit step or one grid instantly at a time duration. However, we can use no movement to restrict the users' velocity and make their movements more realistic. Greenlaw and Kantabutra have modeled the movement of the sources by random walks contained in the set R. Because the sources in the model can move at the different velocities, the walks in R have different lengths. For example, a car can move on by 100 kilometers per hour on a highway, an elephant walking through a forest may move by 1 kilometer per hour, a ship roaming in an ocean can be move by 30 kilometers per hour. Therefore, the model represent the speeds of the sources by natural numbers in the set V. Of course, a given source may not always travel at a constant velocity. It would be worth examining an extension of the model where any source's speed can change over time. Different sources may provide at different wireless signal strength depending on a technology and its battery. Therefore, Greenlaw and

Kantabutra specify c_i to be the coverage diameter of the various sources. This area under the sources' coverage is called the *coverage area*. Because wireless signal can be blocked by some building or other obstacles, the model has a set of obstacles O in a rectangular shape.

Since sources must communicate with each other, Greenlaw and Kantabutra also define the communication protocol used in the model.

Definition 2.5.1 (Coverage Representation) *A coverage of radius c in a two-dimensional grid is represented by the set of grid points within the source coverage and on its boundary.*

Definition 2.5.2 (Overlapping Coverage Area) *Let s, s′ be a coverage or an obstacle in a two-dimensional grid and s ∩ s′ = z. We say that s overlaps s′ if and only if* $|z| \geq 2$. *z is called an overlapping coverage area.*

In [14], the communication protocols are also stated. The sources are always on; they are broadcasting and listening for the communication signal. Users with mobile devices are moving in and out of the range of sources. Mobile devices would like to send and receive messages with one another. Greenlaw and Kantabutra specified the characters that the sources and the users communicate as follows. Let $k > 2$ and $k \in \mathrm{N}$.

- At a given instance in time any two sources with overlapping-coverage areas may communicate with each other in full-duplex fashion as long as the intersection of their overlapping-coverage area is not completely contained inside obstacles. We say that these two sources are **currently in range**. A series $s_1, s_2, ..., s_k$ of sources are said to be currently in range if s_i and s_{i+1} are **currently in range** for $1 \leq i \leq k-1$.

- Two mobile devices cannot communicate directly with one another.

- A mobile device D_1 always communicates with another mobile device D_2 through a source or series of sources as defined next. The mobile devices D_1 at location (x_1, y_1) and D_2 at location (x_2, y_2) **communicate through a single source** *s* located at (x_3, y_3) if at a given instance in time the lines between points (x_1, y_1) and (x_3, y_3) and points (x_2, y_2) and (x_3, y_3) are within the area of coverage of *s*, and do not intersect with any obstacle from *O*. The mobile devices D_1 at location (x_1, y_1) and D_2 at location (x_2, y_2) **communicate through a series of sources** s_1 at location (a_1, b_1), s_2 at location (a_2, b_2),..., and s_k at location (a_k, b_k) that are currently in range if the line between points (x_1, y_1) and (a_1, b_1) is inside s_1's coverage area and does not intersect any obstacle from O and the line between points (x_2, y_2) and (a_k, b_k) is inside s_k's coverage area and does not intersect any obstacle from O.

A Sample Instance of Greenlaw and Kantabutra Mobility Model

An example of a specific instance corresponding to Figure 2.1 is stated. Let $M = (S, D, U, L, R, V, C, O)$, the 8-tuples are defined as follows:

1. Let $S = \{s_1, s_2, s_3, s_4\}$ with initial locations (1,5), (4,5), (5,4), and (4,2) respectively.
2. Let $D = \{000, 001, 010, 101, 110\}$.
3. Let $U = \{u_1, u_2, u_3\}$ with initial locations (2,4), (1,1), and (5,2) respectively.
4. Let $t = 3$ and $L = \{l_1, l_2, l_3\}$, where $l_i = \{000, 000, 000\}$ for $1 \le i \le 3$.
5. Let $R = \{r_1, r_2, r_3, r_4\}$. For clarity Figure 2.1 only illustrates $r_1 = \{101, 001, 101\}$ and omits the other r_i's, which we assume are all (000,000,000), except for r_2 which is twice as long.
6. Let $V = \{1,2,1,1\}$.
7. Let $C = \{2,2,2,4\}$.
8. Let $O = \{(1,1,3,2)\}$.

Initially, there are four sources s_1, s_2, s_3, and s_4 centered at (1,5), (4,5), (5,4), and (4,2), and there are three users u_1, u_2 and u_3 centered at (2,4), (1,1), and (5,2), respectively. A rectangle obstacle has its lower-left corner at (1,1) and the upper-right corner at (3,2).

Sources s_1, s_2, and s_3 each have a coverage with a diameter 2, and s_4 has a coverage with a diameter 4. Sources s_1 are defined its movement by the collection of directions specify in r_1. In this case, s_1 moves south in the first step, east in the second step, and south in the third step. The moves are made with a velocity of $v_1 = 1$, or one grid per a unit of time.

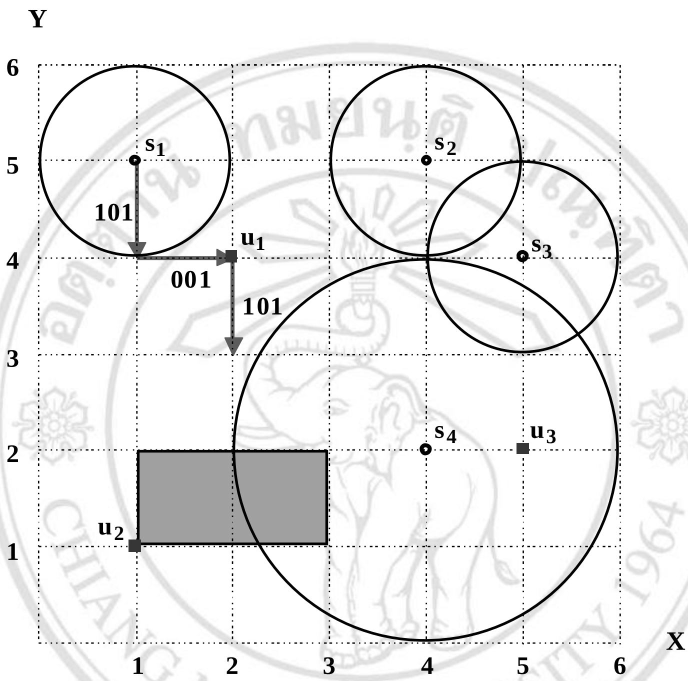

Figure 2.1 Sample instance of the mobility model

Note that initially, for example, sources s_2 and s_3 are currently in range, sources s_2, s_3, and s_4 are a series of sources currently in range, and sources s_1 and s_2 are not currently in range. Initially, users u_1 and u_3 cannot communicate either by a source or a series of sources. After three steps, u_1 can communicate with u_3 through the series of sources s_1 and s_4.

2.6 Growth of Functions

In this section, we review some material on the growth of functions. Much of this review is taken from [18]. Asymptotic notations are used for comparing the efficient of the algorithms. Because of the lower-order term and the multiplicative constant are dominated by the input size when the size is large enough, we compare the efficient of the algorithms from their input size only, *asymptotic* efficiency of algorithms. That means, the algorithm which is asymptotically more efficient is the best choice chosen for solving a problem than others, but for tiny input size.

Asymptotic Notation

The asymptotic notation is the function define the asymptotic running time of the algorithm. It is usually defined by a natural number according to the input size. Basic asymptotic notations are specified as follows.

1. $\theta - notation$ (Big-Theta)

The $\theta - notation$ asymptotically bounds a function from above and below. For a given function $g(n)$, we denote by $\theta(g(n))$ as follows.

$$\theta(g(n)) = \begin{Bmatrix} f(n) : \exists \text{ positive constants } c_1, c_2 \text{ and } n_o \\ \text{such that } 0 \le c_1 g(n) \le f(n) \le c_2 g(n) \text{ for } n \ge n_o \end{Bmatrix}$$

A function $f(n)$ belongs to the set $\theta(g(n))$ if there exist positive constants c_1 and c_2 such that the function $f(n)$ is in between $c_1 g(n)$ and $c_2 g(n)$, for sufficiently large *n*. Because the $\theta(g(n))$ is the set of function, we could write "$f(n) \in \theta(g(n))$". However, we will usually write "$f(n) = \theta(g(n))$" to express the same notion.

2. $\mathrm{O} - notation$ (Big-O)

The $\mathrm{O} - notation$ asymptotically bounds a function from above. We used for an asymptotic upper bound. For a given function $g(n)$, we denote by $\mathrm{O}(g(n))$, as follows.

$$\mathrm{O}(g(n)) = \begin{Bmatrix} f(n) : \exists \text{ positive constants } c \text{ and } n_O \\ \text{such that } 0 \le f(n) \le cg(n) \text{ for } n \ge n_O \end{Bmatrix}$$

A function $f(n)$ belongs to the set $\mathrm{O}(g(n))$ if there exist a positive constant c such that for all values *n* to the right of n_0, the value of the function $f(n)$ is on or below $cg(n)$.

Because the $\mathrm{O}(g(n))$ is the set of functions, we could write "$f(n) \in \mathrm{O}(g(n))$". However, we will usually write "$f(n) = \mathrm{O}(g(n))$" to express the same notion.

3. $\Omega - notation$ (Big-Omega)

The $\Omega - notation$ asymptotically bounds a function from below. We use for an asymptotic lower bound. For a given function $g(n)$, we denote by $\Omega(g(n))$ as follows.

$$\Omega(g(n)) = \left\{ \begin{array}{c} f(n) : \exists \ positive\ constants\ c\ and\ n_o \\ such\ that\ 0 \le cg(n) \le f(n) \ \ for\ n \ge n_o \end{array} \right\}$$

A function $f(n)$ belongs to the set $\Omega(g(n))$ if there exist a positive constant c such that for all values n to the right of n_0, the value of the function $f(n)$ is on or above $cg(n)$. Because the $\Omega(g(n))$ is the set of functions, we could write "$f(n) \in \Omega(g(n))$". However, we will usually write "$f(n) = \Omega(g(n))$" to express the same notion.

2.7 Complexity Theory

The following relevant complexity theory is taken from [18,19,20] and will be used in classifying class of the problems we are interested. Algorithms have variety time complexity on their input size n in the worst cast running time. For example, $O(n^k)$, for a constant k, $O(1)$, $O(\ln n)$, or $O(2^n)$. If an algorithm has worst-case running time equal to $O(n^k)$ for some constant k, we will call it is the *polynomial time algorithms.* However, not all problems can be solved in polynomial time, for example, Halting Problem, SAT, 3SAT, Hamiltonian, Set cover, Clique, and Knapsack problem. Normally, we group the problems that can be solved in polynomial-time as being tractable, or easy, and problems that require superpolynomial time as being intractable, or hard.

"*Problem Class P*" is the problems that can be solved in polynomial time by Deterministic Turing Machine.

"*Problem Class NP"* is the problems that can be solved in polynomial time by Non-Deterministic Turing Machine, *polynomial time verifiable*.

"*Problem Class NP-complete*" is the problem is class *NP* whose status is unknown. No polynomial-time algorithm has yet been discovered for an *NP*-complete problem, nor has anyone yet been able to prove that no polynomial-time algorithm can exist for any one of them. They are the hardest problem in class *NP* since one problem in class *NP*-complete can be solved in polynomial time, all other problems in class *NP*-complete can also be solved in polynomial time.

The theory of *NP*-completeness is designed for *decision problems* because of their counterpart to study in mathematically in the theory of computation. The decision problem composed of generic instances, input to a particular problem, and a question, asked in term of the generic instances. Because we do not know the status of *NP*-complete, we find the way of showing that one problem is no harder or no easier than another. The procedure is called a *polynomial time reduction algorithm*.

Let us consider a decision problem *A* and *B*. A problem *A* can be reduced to another problem *B* if any instance of *A* can be polynomial time rephrase as an instance of *B* and the solution of instance of *B* corresponding to the solution of the instance of *A*. We call *A* has polynomial time reducible to *B* written $A \leq_p B$. And in the sense the problem *A* is "no harder to solve" than the problem *B*. Polynomial-time reductions provides a formal form for showing that one problem is as least as hard as another. That is, if $A \leq_p B$ then *A* is not more than a polynomial-time factor harder than *B*. Because of the *NP*-completeness is to show the hardness of a problem, we use polynomial-time reductions to show that a problem is *NP*-complete as follows.

To prove that a problem Π is *NP*-complete, we have to show that

1. $\Pi \in NP$
2. some known *NP*-complete problem Π' transform to Π.

If Π satisfies only the second properties, we called it is *NP*-hard.

We cannot assume that there is absolutely no polynomial-time algorithm for an *NP*-complete problem because no one prove that there is no polynomial time algorithm for the *NP*-complete problem. And no one know whether *P*=*NP*?

Theorem [18]

If any *NP*-complete problem is polynomial-time solvable, then *P*=*NP*. Equivalently, if any problem in *NP* is not polynomial-time solvable, then no *NP*-complete problem in polynomial-time solvable.

We handle problems with different complexity class in different ways. If a problem is tractable, there is a polynomial time algorithm to solve it. Then, we find asymptotic efficiency and proof correctness of the polynomial time algorithm. We try to improve the effectiveness and proof optimum. For intractable problems, we will find an approximation algorithm to give an approximation value which is close to the optimal values or solve some particular case in polynomial time.

2.8 Graph Theory

In this section, we discuss the basic concepts from graph theory needed in this thesis [21, 22]. The definitions relate to a graph theory are stated as follows:

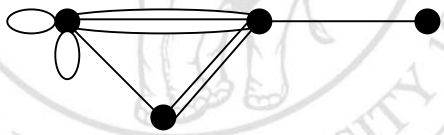

Figure 2.2 Sample graph

Graph is a diagram that can be represented by means of points (vertices, nodes) and lines (edges). We can represent a graph as $G = (V, E)$, where V is the set of vertices and E is the set of edges.

The *degree* of a vertex is the number of edges which have the vertex as an endpoint.

Two vertices u and v of a graph G is said to be *adjacent* if there is an edge to join them. The two vertices are also *incident* to the edge. Two edges are adjacent if they have at least one vertices in common.

If two graphs are a one-one correspondence between their vertices, that is two vertices are joined by an edge in one graph if and only if the corresponding vertices are joined by an edge in the other, we call the two graphs are *isomorphic*. They can be regarded as the same graph.

Simple graphs is the graphs that have no loops or multiple edges If there is more than one edge join a pair of vertices, the edges are called *multiple edges***.** If an edge has two ends from a point, this is called a *loop*.

A *walk* is a sequence of edges. The number of edges in a walk is call *length*. A *path* is the walk which no vertex appears more than once. A *trail* is the walk which no edge appears more than once. A *cycle* is the path that has first and last points are the same. A *circuit* is the trail that has first and last points are the same.

A *forest* is the graph which contain no cycle. A *connected graph* is the graph that has a path connect every two vertices. If there is only one the connected path between every pair of vertices in the graph, we call the graph *tree*. A tree is a connected forest. A *planar graph* is the graph that can be drawn without crossing edge. A *plane graph* is the planar graph that every node can be mapped to a point in a plane.

A *null* graph is the graph with no edges. A *complete graph* is a simple graph which has an edge connect every pair of vertices. A *regular graph* is the graph that every vertex has the same degree.

A *bipartite graph* is the graph that its vertices can be separate into two disjoint sets V_1, V_2 and every edge of the graph G will join the vertex in V_1 to the vertex in V_2. A bipartite graph is *complete* if its edges join every vertex in V_1 to every vertex in V_2. The complete bipartite graph which has $|V_1| = 1$ is called a *star graph*.

A *directed graph* or *digraph* is the graph which the edges have a direction associated to them. The ordered pair represent an edge is called *arc*. An arc (u,v) or uv is an arc from u to v, which is different from arc (v,u) or vu is an arc from v to u.

2.9 Summary

In this chapter, we provide some background on mobility models. There are many wireless models proposed, but we choose the model of Greenlaw and Kantabutra. This model is a theoretical mobility model proposed from the perspective of the complexity theory in a two-dimensional grid. This model also provides framework factors that we commonly use in a mobile wireless network. Those are the reasons that we choose Greenlaw and Kantabutra model. The knowledge about the asymptotic notation, the growth of functions, the complexity theory and the graph theory also states and will be used throughout the contributions that we will show in the later chapters.