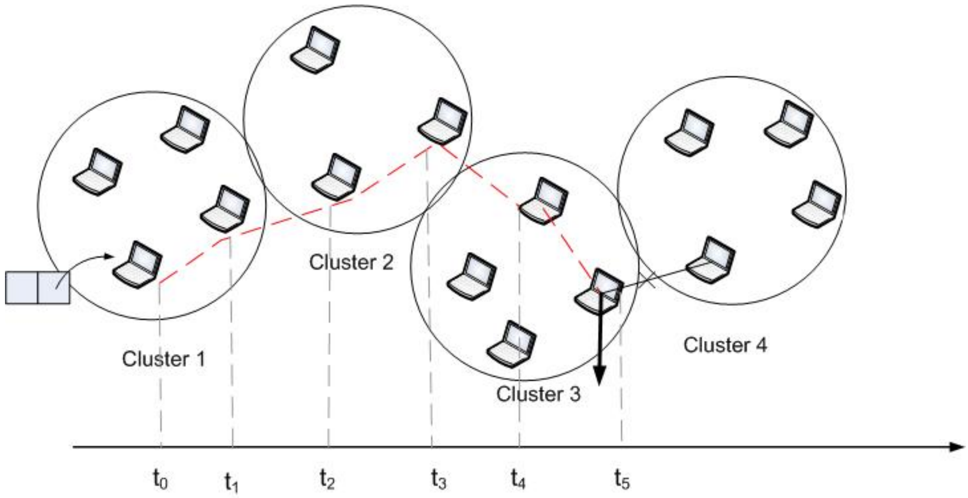
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für
Technische Informatik und
Kommunikationsnetze



Path Length vs Distance in Mobile Wireless Networks

Paraskevi Papoula

Semester Thesis SA-2010-03
November 2009 until March 2010
Advisor: Simon Heimlicher
Supervisor: Prof. Dr. Bernhard Plattner

Acknowledgements

A number of people supported me throughout this thesis in their own unique ways and I would like to express my gratitude to them. I would like to thank Prof. Dr Bernhard Plattner for the supervision of my semester thesis and my advisor Simon Heimlicher. Simon was always there for me, and we had a pleasant collaboration. I am very thankful for his trust in me and unreserved support of my research efforts. I was given the chance to develop my skills in programming and to discover new areas in networking for me. Finally I would like to thank my family for their unconditional emotional support during this period.

Chapter 1

Abstract

Ad hoc networks form arbitrary and dynamic topologies, where there is no central coordination to provide robustness. In such networks, each node acts as a router and a host at the same time, sharing the same channel with the other members of the network. A source communicates with a destination through a multihop route (a sequence of wireless links), in the sense that a packet is forwarded through intermediate nodes until the destination is reached. Each node may need to store an amount of packets in its cache, due to link congestion. Mobility of nodes causes link failures when they move out of each other's vicinity.

The Transmission Control Protocol (TCP) was designed for wired networks, and fails when being used in ad hoc mobile networks. The TCP issues an end-to-end principle according to which forwarding and congestion mechanisms are only running at the end points of the connection. Hence, in case of a link failure at any point of the route the source has to resend the packets.

In this project we examine the performance of intermediate forwarding when the congestion control is distributed in the intermediate nodes of a route. Hence, we study how the taxicab distance, defined as the sum of the absolute differences of the nodes' coordinates, between nodes is related with their hop distance letting nodes move on a lattice $L*L$ with a fixed rate, for a number of time slots. The mobility mode we adopt is the random walk, as it is mathematically feasible to conduct analysis, although it is still too simple to characterize human's real mobility pattern.

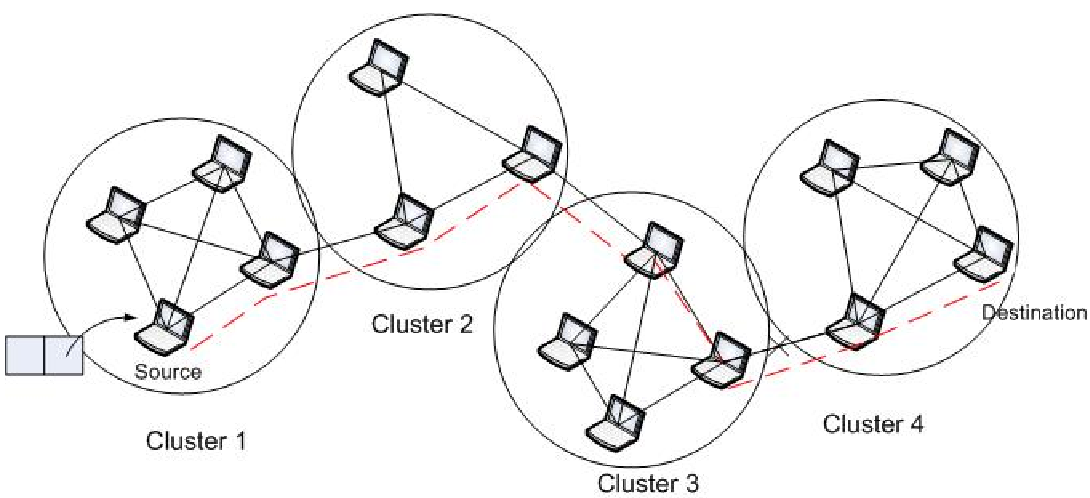


Figure 1.1: Mutltipath route

Contents

1 Abstract **3**

2 Introduction **9**
2.1 Goal of the Project 11
2.2 Related Work 11
2.3 Thesis Outline 12

3 Background Information **13**
3.1 Floyd-Warshall Algorithm 13
3.2 Monte Carlo Simulations 14
3.3 Hop Count 14
3.4 Taxicab Distance 15
3.5 Coverage and Node Degree 15
3.6 Approach 17

4 Implementation **19**
4.1 Conditional Probability of hop count given hop count 19
4.2 Conditional Probability of taxicab distance given taxicab distance . 21
4.3 Conditional Probability of taxicab distance given hop count 22
4.4 Conditional Probability of hop count given taxicab distance 23
4.5 Taxicab Analysis 26
4.5.1 State Probabilities 26
4.5.2 Transition Probabilities 29
4.6 Hop count Analysis 32

5 Conclusion **35**
5.1 Conclusion 35
5.2 Future Work 35

6 Appendix **37**

References **47**

List of Figures

1.1 Mutltipath route 3

2.1 Link failure 9
2.2 Taxicab Distance 10

3.1 Three possible positions of nodes in the grid 16
3.2 Topology plot N=1000, L=400 18

4.1 Hop count distribution, initial hop count 3 20
4.2 Hop count distribution, initial hop count 2 20
4.3 Lattice with two nodes in odd hopcount 21
4.4 Taxicab distribution, initial taxicab distance 0 22
4.5 Taxicab distribution, initial hop count 4 23
4.6 Taxicab distribution, initial hop count 1 24
4.7 Hop count distribution, initial taxicab distance 1 25
4.8 Hop count distribution, initial taxicab distance 2 25
4.9 Markov chain for taxicab distance 26
4.10 Even subset of states for taxicab distance 27
4.11 Odd subset of states for taxicab distance 28
4.12 Markov chain for hop count metric 33
4.13 Two hop neighbors separated by a known distance x as depicted in [1] 34

6.1 Hop count distribution, initial hop count 2 37
6.2 Hop count distribution, initial hop count 3 38
6.3 Hop count distribution, initial hop count 5 38
6.4 Taxicab distribution, initial taxicab distance 0 39
6.5 Taxicab distribution, initial taxicab distance 1 39
6.6 Hop count distribution, initial taxicab distance 5 40
6.7 Hop count distribution, initial taxicab distance 6 40
6.8 Taxicab distribution, initial hop count 1 41
6.9 Taxicab distribution, initial hop count 2 41
6.10 Taxicab distribution, initial hop count 3 42
6.11 Taxicab distribution, initial hop count 4 42
6.12 Taxicab distribution, initial hop count 5 43
6.13 Taxicab distribution, initial hop count 6 43
6.14 Hop count distribution, initial taxicab distance 1 44
6.15 Hop count distribution, initial taxicab distance 2 44
6.16 Hop count distribution, initial taxicab distance 3 45
6.17 Hop count distribution, initial taxicab distance 4 45

Chapter 2

Introduction

Transport Control Protocol (TCP) was designed for wired networks, to provide reliable data transmission while controlling data flow and avoiding network congestion. TCP also controls the network traffic and the rate at which data is exchanged. It is a connection oriented protocol, which uses a number of mechanisms to control the congestion, such as acknowledgements, slow start and retransmission timeout. It uses a three-way handshake to establish a connection between the pair source-destination, where the packets are forwarded through intermediate nodes.

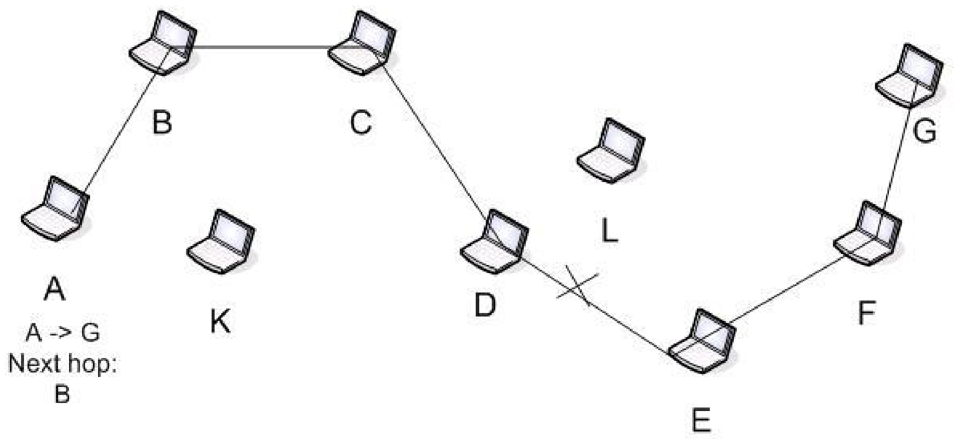


Figure 2.1: Link failure

If the source does not get an acknowledgement from the destination G within a time interval it sends the same data again. When the source retransmits the packets usually it is more hops away from the destination than any intermediate node. A solution to that problem could be to distribute the transport control to the intermediate nodes so that they are responsible for delivering the packets to the destination. To make sure that this is a solution in the following chapters we observe the relation between the taxicab distance of two nodes and their hop distance and introduce the intermediate forwarding. As shown in figure 2.2 the taxicab distance between Node A and Node B is the orange line connecting them. The name of this metric is established by the taxicab drivers in the streets of Manhattan. Intermediate Forwarding (IF) as will be explained in the next section is our suggested protocol where the nodes of a multihop path transfer data hop by hop and in case of a link failure the node prior to it will retransmit the packet. For example in figure 2.1 where the source node A sends packets to the final destination G, according to IF if the link between nodes D and E fails then node D will have to retransmit the packet.

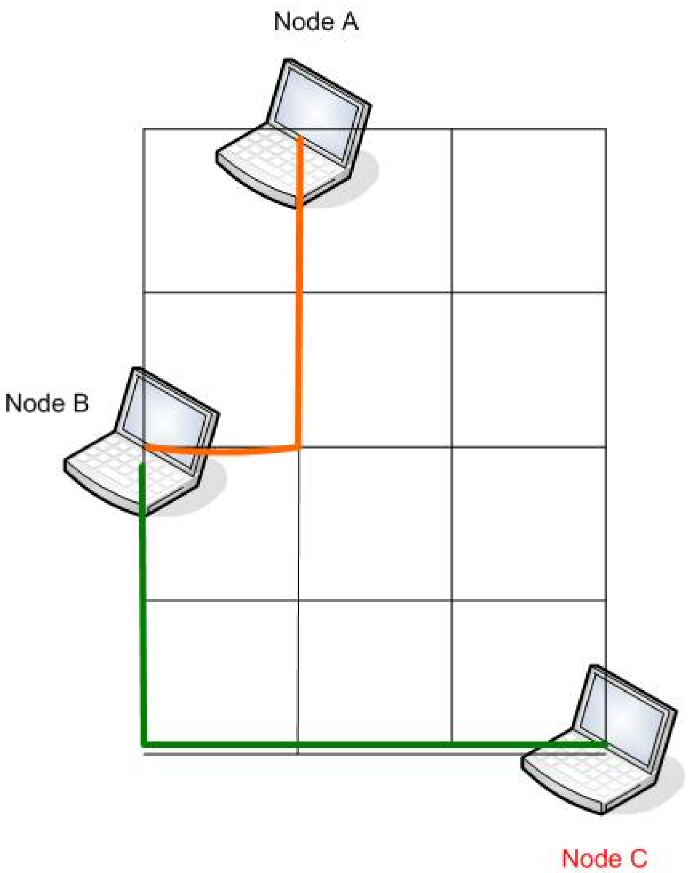


Figure 2.2: Taxicab Distance

When we look at mobile ad hoc networks (MANETs) we may assume that their members are constantly connected, partially connected or disconnected. It is plain to understand that the first and last category are affected by the density of the network. In real life however, partially connected networks [2] are easily formed due to the mobility of the nodes. This kind of networks are formed when the source and the destination are not joined directly but instead different nodes complement the *broken* route providing multi-hop paths until the destination is reached. These multi-hop paths allow transmitting data closer to the destination.

In partially connected graphs the network is divided into groups which are not connected and each node is only aware of nodes within its cluster. This means that if the source and the destination are in different clusters they either have to wait until they are in the same cluster, or a node might move to the desired cluster and forward the data packets to the destination. The latter is not enough though as according to the TCP protocol the source demands acknowledgement from the destination within a timeout limit, so not only the data packets have to go through, but also the acknowledgement packets otherwise the source will resend the data.

What we study in this project is the case where the data is forwarded from an intermediate node D which acts as the source. Assuming that all nodes are constantly moving, we calculate the probability that after some time a node which was initially closer (either in means of distance, or of hops) to the destination will still be closer to it than the source which initially was further away. The data are forwarded form the intermediate node only if there is a different valid route in its cache memory, thus we don't need to store them for long periods of time. The end-to-end acknowledgement still has to go through the partially connected network from the destination towards the source as is the requirement for a reliable service.

2.1 Goal of the Project

In this project we address the TCP layer of the TCP/IP suite, where we have observed problems like time wasted for route establishment, and retransmissions in absence of acknowledgement. We focus on two different forwarding schemes; the source forwarding and the intermediate forwarding. In order to evaluate the two mechanisms we use two metrics, one for the physical distance between nodes and one for the hops between the nodes. We use Markov chains to model the two metrics and perform simulations in a network with mobile nodes.

Source Forwarding According to this mechanism the source forwards packets along an established route towards destination. This path may have been established based on a variety of criteria such as shortest path, lower energy, less traffic etc. ACKs are needed for reliability. The main reasons for a poor performance are route changes or failures and packet losses. In case of a broken link, the node prior to the link failure generates route failure messages and notifies the source to retransmit the data packets. Source forwarding performs well in networks where no or a few packets are lost. However in case of an ACK absence the data packet will be retransmitted, even if it has been received. Thus there might exist multiple copies of the same packet along the network wasting resources.

Intermediate Forwarding In this mechanism each node of a multihop route is responsible for congestion and link-flow control. At the same time there is an end-to-end service operating on the end points which performs global congestion control and provides reliable data delivery. To be more specific, the node that currently has the packet decides which of its neighbors should receive the packet based on information from the routing protocol. The data packets are transferred hop by hop and they are temporarily stored in each node. Doing so, there is an effort to fool the end points that no packet is lost, to avoid broadcast of route error messages and data retransmission. In case of a packet loss the sending node (of that broken link, and not the source) will retransmit the packet. During a route failure our suggested protocol does not get affected, since invalid routes can still be used as long as the next hop is up. The three fundamental tasks of a transport protocol, namely the end-to-end reliability, flow control and congestion avoidance, are preserved as described in [3]. Throughout the project we assume that once the packet is sent from one node it will be received successfully, ignoring collisions, interference and other lower layers' errors.

2.2 Related Work

MANETs have been addressed in several works. [4] presents a formal model to predict the lifetime of a routing path based on the random walk model. Nodes are roamed in a hexagonal cellular system. A routing path is regarded as a sequence of vectors (*states*), each representing one wireless link. The sequence of states is then considered a stochastic process. In [5] statistical techniques are used to derive distance estimates between nodes without using any sort of distance measuring hardware. The relationship between the density of the nodes, the number of hops separating two nodes, and the distance between these two nodes is investigated, for one and two dimensions. Nodes are placed according to a planar Poisson process, and there is no mobility in this model. A closed formula is provided for the one dimension, and a recursive method yielding an approximation for the two dimension. In [6] the length of a two dimensional random walk with non uniform step length is derived and applied to a cinephotomicrographic analysis of human cell movement. The movement is first observed in absence, and then in presence of external forces. In [7] an approach to support random walks that uniquely converge to application desired node visitation probability distributions is investigated. The random walks of Zong et al. are built on the Metropolis-Hastings algorithm. They have derived analytical bounds for the convergence time on P2P networks, proven that for the convergence-guaranteed random walks the only required state at each node consists of the network degree and visitation probability of its direct neighbors and then they presented the effectiveness of random walks in realistic applications. Alagar in [8] shows how the distribution between random points can be found when the points are chosen uniformly and independently in a hypersphere or in two adjacent unit squares, by using the Crofton technique. In MANETs frequent path changes due to node mobility, wireless propagation effects etc result to partially connected networks. [2] and [9] study the intermediate forwarding vs the source forwarding when there is no end to end connectivity. Assuming that an alternate route is available immediately when the end-to-end path fails, and if the length of the alternate route is stochastically monotonic in the length of the primary route, then intermediate forwarding stochastically dominates end-to-end forwarding. [10] introduces a stochastic model to characterize the delay incurred by message in a mobile ad hoc network. The probability distribution function is obtained for the number of copies of the message at the time the message is delivered for two protocols: the two hop and the unrestricted multicopy protocol. In [1] a closed form of equation is provided for the computation of the probability that two nodes initially being in distance x are also k- hop neighbors. Nodes in this model are uniformly distributed in a square area and are not mobile. We follow the same reasoning, but apply it in a grid topology where nodes perform random walk.

2.3 Thesis Outline

In the next chapter we present the background information that the reader needs in order to understand the Floyd-Warshall Algorithm, and the metrics used for our model (taxicab distance and hopcount). The fourth chapter describes the model analysis and the implementation. Finally in the last chapter we present the results and we conclude the thesis.

Chapter 3

Background Information

In this chapter we firstly present information about the Floyd-Warshall algorithm we use to calculate the shortest path and the monte carlo simulations. Furthermore we describe the two metrics we use in our network to evaluate the two forwarding mechanisms, along with some information about the coverage and the node degree of each node. Finally we explain our approach of the problem presenting the four different distributions that we are interested in.

3.1 Floyd-Warshall Algorithm

Each node in our network maintains an adjacency matrix. We need an efficient Breadth First Search algorithm to provide as with the shortest paths from each node to every other node and thus we chose Floyd Warshall algorithm. An advantage of the Floyd-Warshall algorithm is its simplicity, which is particularly appealing if the input graph is represented by an adjacency matrix. A consequence of this simplicity is a small constant factor hidden in the big-Theta of the $\Theta(n^3)$ running time where n is the number of vertices in the graph. A big advantage the Floyd-Warshall algorithm has over the Breadth First Search based algorithm is that it also works for weighted graphs. In comparison to Dijkstra (Depth First Search), which only gives us the shortest path from one source to the targets, Floyd-Warshall gives us the shortest paths from all sources to all target nodes. Other uses for Floyd-Warshall are to find connectivity in a graph (known as the Transitive Closure of a graph). After running the algorithm on the adjacency matrix, the element at adj[i][j] represents the length of the shortest path from node i to node j. Below we present the pseudo-code for the algorithm:

```
for (k = 1 to n)
   for (i = 1 to n)
      for (j = 1 to n)
            adj[i][j] = min(adj[i][j], adj[i][k] + adj[k][j]);
      end
   end
end
```

[4] argues that if packets are routed along shortest paths from source to destination robustness is weakened, because each node is crucial and cannot easily be replaced in case it moves out or the vicinity of its neighbors. However, we still use the shortest path routing because it is a well known and well defined metric that is easy to use. By allowing multiple nodes to be at the exact same place on the grid, we decrease the importance of each node, but the problem mentioned in [4] remains.

3.2 Monte Carlo Simulations

Numerical methods that are known as Monte Carlo methods can be described as statistical simulation methods, where statistical simulation is defined in quite general terms to be any method that utilizes sequences of random numbers to perform the simulation [11]. As mentioned in [12] (last checked on 26th of July 2010) in many applications of Monte Carlo, the physical process is simulated directly, and there is no need to even write down the differential equations that describe the behavior of the system. The only requirement is that the physical (or mathematical) system be described by probability density functions (PDFs). Once the PDFs of the system are known, the Monte Carlo simulation can proceed by random sampling from the PDFs. Many simulations are then performed (multiple *trials* or *histories*) and the desired result is taken as an average over the number of observations (which may be a single observation or perhaps millions of observations). In many practical applications, one can predict the statistical error (the *variance*) in this average result, and hence an estimate of the number of Monte Carlo trials that are needed to achieve a given error.

Below we present briefly the major components of a Monte Carlo method. These components comprise the foundation of most Monte Carlo applications. An understanding of these major components provides a sound foundation to construct our own Monte Carlo method. The primary components of a Monte Carlo simulation method include the following:

- Probability distribution functions (PDFs)given that the physical (or mathematical) system must be described by a set of PDFs.
- Random number generator a source of random numbers uniformly distributed on the unit interval must be available.
- Sampling rule a prescription for sampling from the specified PDFs, assuming the availability of random numbers on the unit interval, must be given.
- Scoring (or tallying); the outcomes must be accumulated into overall tallies or scores for the quantities of interest.
- Error estimation; an estimate of the statistical error (variance) as a function of the number of trials and other quantities must be determined.
- Variance reduction techniques; methods for reducing the variance in the estimated solution to reduce the computational time for Monte Carlo simulation.

3.3 Hop Count

In our approach, we chose to analyze how the hop count and the distance of two nodes are related. Thus the metrics used are the hop count and the taxicab distance of every pair of nodes.

When two nodes of an ad hoc network try to communicate, their messages have to be relayed from a number of intermediate nodes. Depending on the application and its needs, the chosen path should fulfill different requirements, such as minimum energy consumed, minimum interference, minimum delay, etc. The Floyd-Warshall algorithm that we used calculates the shortest path between two nodes after every time slot. Given that the nodes move at every time slot the path can vary between consecutive slots as the algorithm always finds the best possible route. The number of hops from the source node to the destination is referred to as *hopcount*. If the source is at the exact same position with the destination, then their hop distance is one, because the message has to be transmitted once over the wireless channel to reach the destination.

3.4 Taxicab Distance

The latter metric used, is the taxicab distance between nodes. The Taxicab Geometry is used by taxicabs in an ideal city with a rectangular grid of streets. Distance is measured not as a virtual line connecting two points, but as a taxicab drives. Hence the distance between two locations is the sum of the vertical and the horizontal distances between them, rather than the Euclidean distance given by the Pythagorean Theorem. In contrast to the hop count metric, if two nodes are at the exact same position, their taxicab distance is zero. It is irrelevant whether there are intermediate nodes between source and destination pair. Since each node has only four options for its next position, the taxicab distance between two nodes is discrete and easy to calculate. Supposing that their initial distance is d and given that each node moves one grid unit per slot, towards one of the four directions of the grid, then their distance after one time slot can get one of the following values with certain probability:

$$d(t+1) = \begin{cases} d(t) + 2 \\ d(t) \\ d(t) - 2 \end{cases}$$

Note that each possible option is not equiprobable and depends on whether the nodes are in the same line or not, as we will show in the Markov chain analysis that follows.

3.5 Coverage and Node Degree

Coverage is defined as the number of times that the area is covered by all nodes of the network. Each node can cover a circle area, centered at its position, with radius R. Hence, in case of an internal node (not close to the borders of the grid, such as B in fig. 3.1) the area covered is equal to

$$E_{\text{internal}} = \pi \cdot R^2 \tag{3.1}$$

while, in case of a border node (such as A or C), the area depends on its exact position, subtracting the circular sector that is out of our grid as depicted in figure 3.1. A node that its area exceeds two borders of the grid, is considered a special case, and handled accordingly. Our algorithm can be used for nodes that take any arbitrary position in an area, and not only in the case of a grid. The coverage is given by the formula

$$C = \frac{(N \cdot \pi \cdot R^2)}{L^2} \tag{3.2}$$

and is in accordance with the results from simulations. It takes positive values, and a value greater than *1* means that the area covered by the nodes overlaps.

Node degree is related with the coverage ($NodeDegree \approx Coverage - 1$), because if each node has k neighbors in average, then this means that the part of the area covered by these k nodes overlaps, and thus it is counted k times. If this happens with every node, then the whole area will be covered k times. The results from the simulations match the theoretical ones with a small deviation of order *1%*, due to the border effect. But when the node degree of only the internal nodes is measured, then it matches with the formula. Nodes in the border don't have the

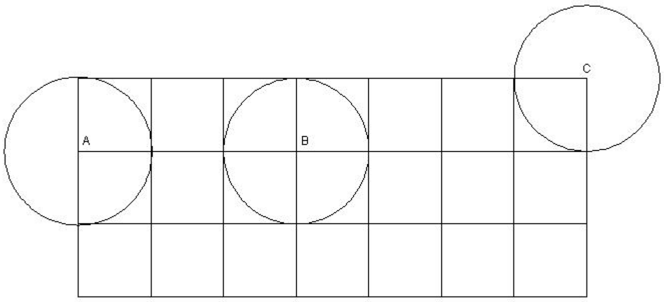


Figure 3.1: Three possible positions of nodes in the grid

same neighbors as the internal ones, due to the smaller area that they cover. Thus, their node degree is smaller than it would be if they were internal, decreasing the average node degree. A simple rule-of-thumb formula for the node degree is

$$D = \frac{((N-1) \cdot \pi \cdot R^2)}{L^2} \tag{3.3}$$

3.6 Approach

To study the problem described in previous section, we focus on a network consisting of N nodes, uniformly distributed in an area $L*L$. If the nodes are less than R distance away, they are considered to be neighbors or otherwise stated, their transmission and reception range is a circle centered in their position with radius R meters. The nodes are allowed to move on a grid, with a fixed step towards any of the four directions (equiprobable) for a number of timeslots T. Given any vertex of the grid which represents the current location of a host, the host will move to the four neighboring vertices in the next time unit with equal probabilities *(1/4)*. This describes our discrete time random walk model. In the case of the border nodes, if they choose to move out of the limit of the area, they are reflected towards the opposite direction.

Nodes communicate with each other either directly, if one is in the vicinity of the other, or through other nodes, forming a multihop path. The intermediate nodes forward the packets to the node that is closer to the destination.

To make sure that nodes remain uniformly distributed at each time slot, we observed that each node was moving almost the same number of times towards all four possible directions, and that similar number of nodes have been to each coordinate (x and y). We use Floyd Warshall algorithm to find the shortest path between source and destination, which also provides us the next node in that path that will forward the packets. A single execution of the algorithm will find the shortest paths between all pairs of vertices.

We perform simulations in Matlab version R2008a, varying the value of N, L, R and we observe the average node degree over time, the coverage of the area, and we calculate four types of probabilities. In each time slot, the coordinates of each node are updated, creating a new topology.

Below we describe the four probabilities that we are interested in:

Hop Count Given Hop Count At first, for all the k-hop neighbors in the initial topology we observe how many hops away they are, after one, two, three, up to 5 time slots. In the appendix (6.1, 6.2, 6.3) the bar chart depicts the probability of k-hop connectivity among the nodes that were two-hop neighbors in the previous time slot. Since there are N nodes in the topology, two nodes would be disconnected if they needed more than N hops to reach each other. Since most of the probable hop numbers are lower than ten, the bar chart would not be explicit if the x-axis maximum was N. Hence, we shifted this number of hops, to a number of hops with zero probability and more specifically in this figure 0-hop neighbors represent the disconnected nodes.

Taxicab Distance Given Taxicab Distance Furthermore, we observe how the taxicab distance between nodes changes after one time slot. Namely, we randomly choose a pair of nodes that have a taxicab distance of h and measure their distance. Then, by monitoring them and every other pair of nodes that are at the same distance over time, we find the possible Taxicab distances that these nodes can have in the next time slot, and the percentage of nodes that have each possible distance. In 6.4, 6.5, 6.6, 6.7, the x-coordinate of each point represents the distance that was observed, and the y-coordinate, how likely it is to happen. This is a taxicab probability for the next time slot, conditioned on the taxicab distance of the current time slot.

Taxicab Distance Given Hop Count As a next step, with the two following probabilities we try to find how the number of hops and the taxicab distance of a pair of nodes are related. Thus, we find all the pairs of the topology that are k-hop neighbors and measure their taxicab distance after one time slot. In the produced figures 6.8 and 6.13 in the appendix, the x-coordinate of each point represents the possible taxicab distance between two nodes that were previously k-hop neighbors, and the y-coordinate is the probability corresponding to that distance. This is a taxicab probability for the next time slot, conditioned on hopcount of the current time slot.

Hop Count Given Taxicab Distance For the last probability of this section, we find nodes in our topology that are a specific taxicab distance away from each other at the current time slot, and check how many hops away they are at the next time slot. This probability is depicted in figures 6.14 and 6.17. This is a hopcount probability for the next time slot, conditioned on the current time slot.

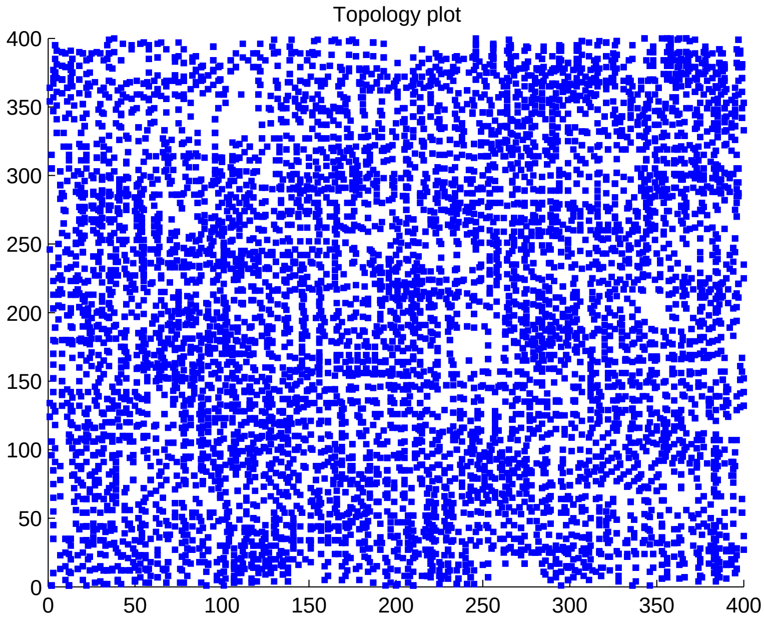


Figure 3.2: Topology plot N=1000, L=400

Chapter 4

Implementation

In this chapter we will explain in more details how we computed the aforementioned conditional probabilities, and how we evaluate the results. In our topology we have N nodes that constantly move in the grid with a step of one grid unit per time. The range of each node is one grid unit. We are interested in all the pairs of nodes that at the current time slot fulfill a condition of either hop count or taxicab distance.

4.1 Conditional Probability of hop count given hop count

The first distribution of hop count is formed from pairs that have a specific hop count at the current time slot. Assuming that we want to find the distribution of hop count between pairs that currently have one intermediate node, we will go through all the pairs in the topology, choose the ones that have one intermediate node, and then observe how this count changes in the next time slot. If we keep track of them for more than one time slot, then we end up with the subplots depicted below in figure 4.1. Here we need to note that we don't scan our topology only once at the beginning of the simulation for pairs to match the requirements, but instead, after each simulation round, we perform that check again and if pairs match the condition of the hop count, they are also observed over time. The data gathered are stored in a three dimensional table (*hop count*), because we need to have indices for the time slot, the current hop count and the next hop count. Thus, if *hop count(t+3,h,h')=K* then this means that *3* time slots after the beginning of the simulation, there are K per cent of the pairs fulfilling the condition that have hop count *h'* and that at time t they had hop count h.

In order to have an explicit chart, we have shifted the disconnected nodes to a value that can never have non zero probability. Thus the bar at *x=-1* represents the nodes that there is no path between them.

In figure 4.1 we present the probability *Pr(h'|h=3)*, for 5 consecutive time slots, one for each subplot. Hence the first subplot depicts the distribution after one time slot, the second one refers to the distribution as it is after two time slots and so on. As we can see in figure, when looking at the first subplot the percentage of pairs that are at hop count *5* at the next time slot, given the fact that they were initially at hop count *3*, is approximately *25%*.

Another fact that we observe is that there are only odd values in this figure. This makes sense if we consider that given an odd number as the initial hop count, we can find another hop count for the new position of the nodes to form a loop. This loop can only have an even number of edges and it is the sum of the initial hop count plus the new value of the hop count. Thus, if we know that the initial one is

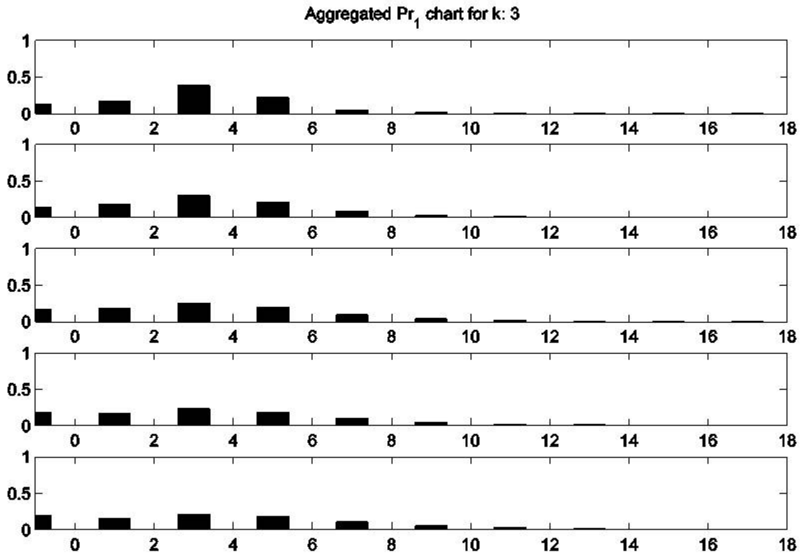


Figure 4.1: Hop count distribution, initial hop count 3

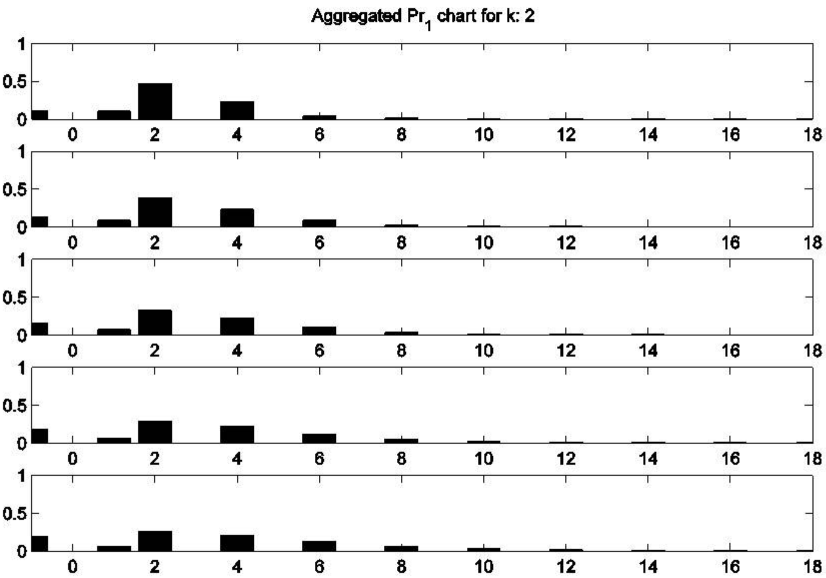


Figure 4.2: Hop count distribution, initial hop count 2

odd then the next one will have to be odd again. Similar is the case with an even initial value, as the sum of two even numbers remains even. This can be verified by figure 4.2. Another reason that justifies the existence of only odd values in the figure is the lattice. For example, lets observe figure 4.3. Nodes A and B have an initial hopcount shown by the red line of 3 hops. In the next time slot, assuming that they are still connected, the set of possible hop counts can be the following: 1, 3, 5, 7,..., 2L-1. These values are obtained by examining the different combinations of the nodes' next positions.

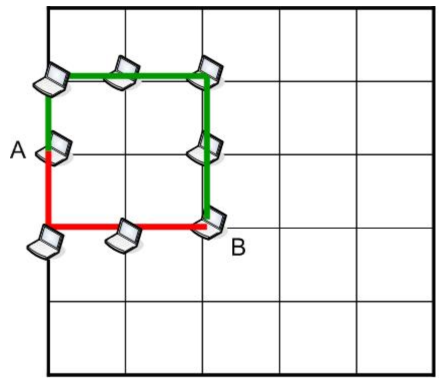


Figure 4.3: Lattice with two nodes in odd hopcount

For example nodes A and B will be only one hop count away in the case that node A moves right (or down) and node B up or left. Higher hop count values are also possible because nodes might be connected over a longer path, such as the green line for example. Even in this case the number of hops will still be odd because the produced path will always consist of the direct line between A and B increased by 2 hops, 4, 6, 8, ... 2*k, where k is any value in [1,L).

Another point we need to notice here, is the fact that when two nodes are located at the exact same position they are still one hop away. In figure 4.2 we can see that although the initial value is even namely *2*, then the value of *1* hop count is also possible despite being an odd number. This happens due to the fact that if two nodes are initially at a hop count of two (which means they are at a maximum taxicab distance of two), then in the next time slot they can be at the same position. But as stated before two nodes at the same position have hop count equal to 1. This option remains in the next time slots as we can see in the same figure.

More bar charts can be found in the appendix.

As we observe in the bar charts, the most likely value of the hop count between pairs in the next time slots, remains the value of hop count that they initially had.

This means that if a node was initially close in terms of hops to another one, then it is more likely that it will remain close. Otherwise, the next more likely option is the greater value for the hop count as we can see in 4.2 and 4.1.

For larger initial paths ($h>5$), it is obvious in the relevant figures in the appendix, that it is more likely to have a link failure after a few time slots.

4.2 Conditional Probability of taxicab distance given taxicab distance

Here, we examine the physical distance of the nodes. The metric we use is the taxicab distance, namely the sum of the difference between the coordinates of the two nodes ($|x_1 - x_2| + |y_1 - y_2|$). For example if the coordinates of two nodes are *(3,1)* and *(4,10)* then their distance is *1+9=10*. In this case, if two nodes are at the same position then their taxicab distance is *0*, in contrast to the hop count which cannot take this value. The condition we observe here, is the initial taxicab distance of nodes. Assuming we are interested in nodes that have initial taxicab distance of *2* units we follow the procedure described in the previous subsection and get the distribution of pairs for the next time slots.

If the initial taxicab distance is even (odd), it will remain even (odd). This happens due to the fact that both nodes move by one step at every time slot, thus increasing or decreasing their distance by zero units if they move towards the same direction, or by two units in any other case. This also explains the fact that at the first subplot we will always observe only three possible values for the distance between two nodes (apart from the case where nodes are at the same position, thus their distance cannot decrease). The data is stored again in similar table, with indices for the current distance, the future one, and the time slot. Some results are shown in 4.4. The maximum value of the taxicab distance between two nodes is $L+L$.

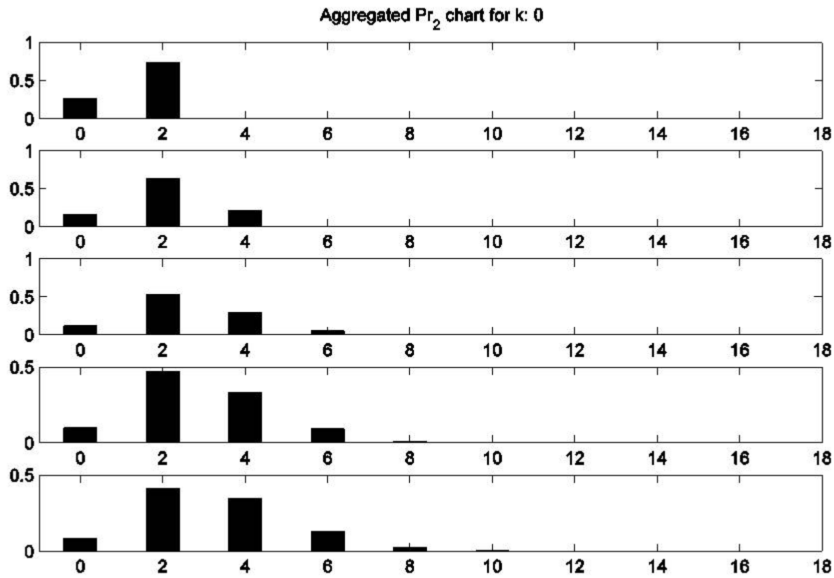


Figure 4.4: Taxicab distribution, initial taxicab distance 0

As we observe in the figure for two nodes that are initially at the same position, it is more likely that they will be further away after a few time slots, namely at a taxicab distance of *2* or *4* units after *5* time slots.

4.3 Conditional Probability of taxicab distance given hop count

In this section we observe the distribution of the pairs that will be at a taxicab distance conditioning on hop count in the current time slot. Hence, we search for the pairs that are at hop count h in the current time slot, and then we monitor their taxicab distance during the next few time slots, as we did in the two previous sections.

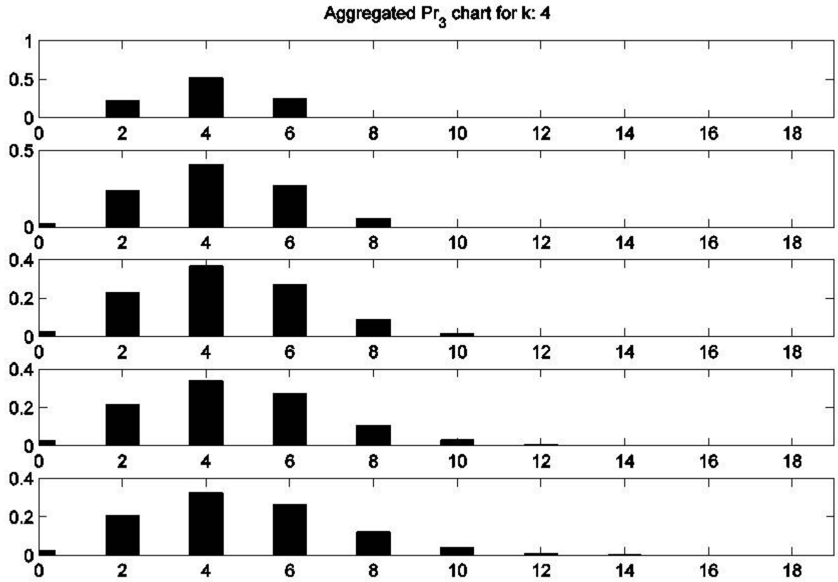


Figure 4.5: Taxicab distribution, initial hop count 4

Figure 4.5 depicts the distribution of taxicab distance of pairs in the next time slots, conditioning on their current hop count which is 4. From this figure we derive the information that it is more likely that nodes will have a taxicab distance of 4 in the next time slots. Floyd Warshall provides the shortest path between nodes. This is at the same time the maximum taxicab distance between these nodes. Thus the information that two nodes are currently 4 hops away, reveals that their maximum distance in the next time slot can be 6, as we can confirm from the 4.5. This also explains why we observe only even possible taxicab distances given that the two nodes are an even number of hops away. In the last sub-chart, after 5 time slots, we notice that the distribution is smoother than the previous time slot.

When nodes are one hop away, they can either be at the same position having zero taxicab distance, or have a taxicab distance of one. We can observe this in figure 4.6. First of all, one time slot later, there are no possible values larger that 3. Then, we have two *cases* in the same chart. One case consists of the even distances 0 and 2 and the other one of the odd distances 1 and 3. In our topology, it seems more likely for nodes to be at different positions in the next time slot given that their hop count is one now. If we observe the chart in the next time slots, it is clear that the bars with higher probability belong to the case that the nodes are not at the same position initially.

4.4 Conditional Probability of hop count given taxicab distance

The last distribution that we observe concerns the hop count of pairs that were initially at a given taxicab distance. In this case if the nodes are at the same

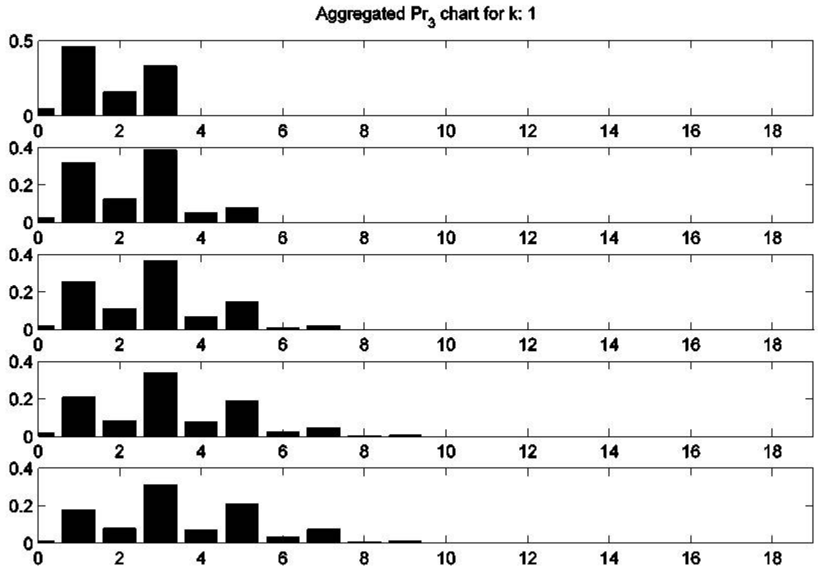


Figure 4.6: Taxicab distribution, initial hop count 1

position at *t=0*, their hop count is one, and at *t=1* their hop count can take every possible even value up to the number of nodes in our topology. The reason for this result is that the taxicab distance provides the minimum number of hops between two nodes, but the upper limit for the possible values of the hop count is in the order of the number of nodes. For example if two nodes are at taxicab distance of two, packets will be forwarded at least two times until they reach the destination. For the worst case we can say that there might exist a path consisting of *N-1* nodes until the destination is reached. Otherwise the nodes will be disconnected. In figure 4.7 we see that the pairs that are at taxicab distance of one unit, will most probable be at one hop count away in the next time slot, but in the next time slots, it becomes more likely that the path between them will grow up to 3 hops. The possible values of the hop count for nodes that are at a taxicab distance of 3 can be larger than 17, yielding the volatility of the network.

The bar at the negative part of the x-axis represents the percentage of the disconnected pairs. From figure 4.7 is obvious that while two nodes are only one grid unit away now and communicating directly, in the next time slot it is likely that they lose contact.

Another thing we should note is that if nodes are at taxicab distance 2, the allowed values of hop count for the time following slots should remain even, for the same reason that we explained in the previous sections. However, in figure 4.8 we see that an odd value has non zero probability. This is due to the fact that nodes at the same position are one hop away.

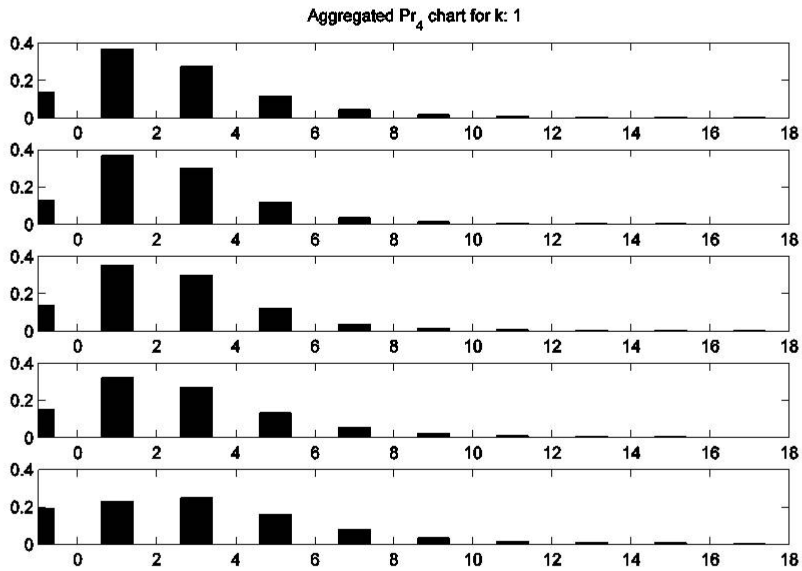


Figure 4.7: Hop count distribution, initial taxicab distance 1

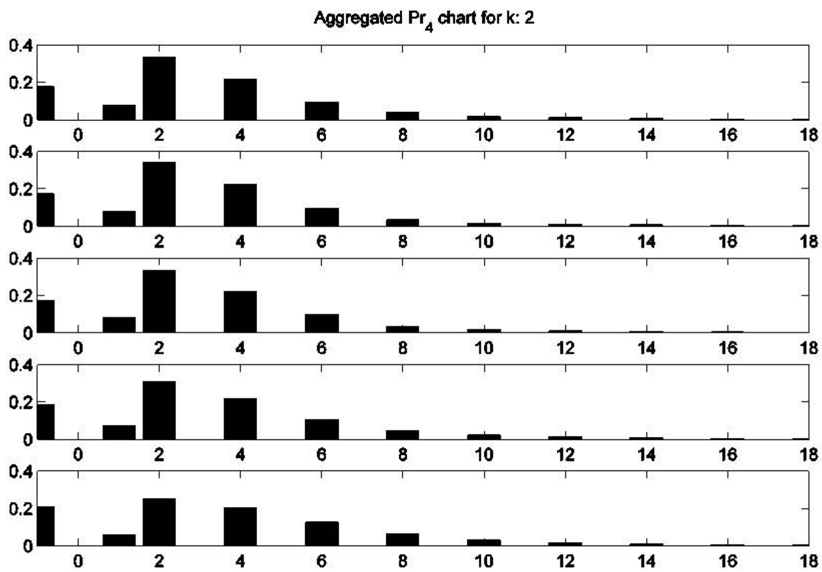


Figure 4.8: Hop count distribution, initial taxicab distance 2

4.5 Taxicab Analysis

In our model we used the taxicab metric to measure the distance between node pairs. Let's assume we have two nodes that move constantly according to our mobility model. Then their distance changes over time with the property that the value of it in the next time slot depends only upon the value that the distance has in the current time slot. In other words there is no memory in the way that the taxicab metric denoted as D in 4.16 evolves over time. This is also described by the formula

$$Pr[D(t+1)|D(t), D(t-1), ..., D(1)] = Pr[D(t+1)|D(t)] \tag{4.1}$$

Since the node distribution is uniform, and remains uniform during all time slots of a random walk, the difference of the coordinates $(Y_1 - Y_2, X_1 - X_2)$ is also uniform over time [13]. Having in mind these two properties i.e. no memory over time and uniform distribution of nodes, we use Markov Theory to model this metric. In the chain of figure 4.9, each state represents the taxicab distance between any pair of nodes. The states are connected if the transition probability between them is not zero.

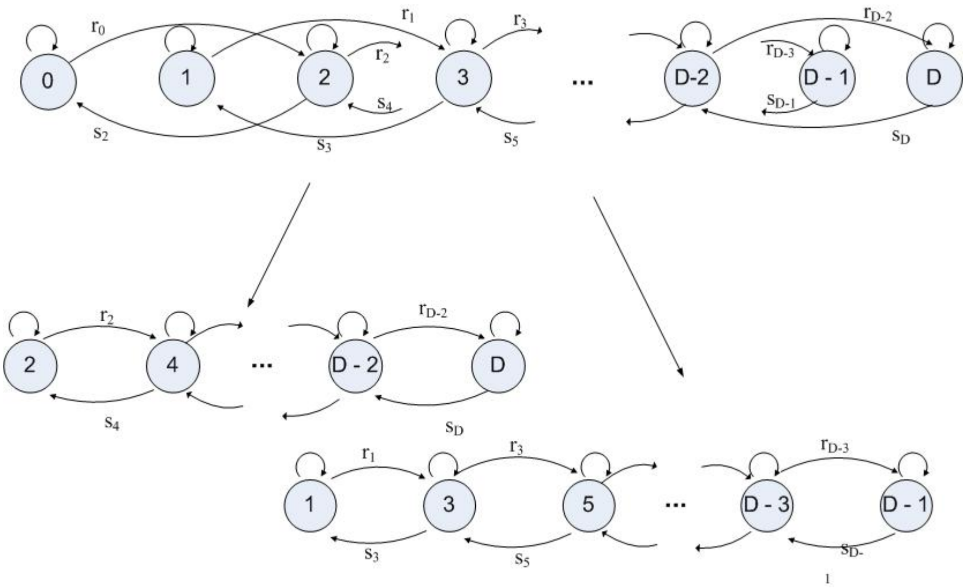


Figure 4.9: Markov chain for taxicab distance

We observe that when the taxicab distance between two nodes is *even* in the current time slot, then its next value can only be *even* as well, and when its initial value is *odd*, then the next value will be only *odd*. This is explained by taxicab geometry, the mobility model that we have adopted. Both nodes move every time slot, which means that the difference in the x- and y- coordinates of a pair of nodes if changed, will be increased or decreased by two units.

4.5.1 State Probabilities

Since every odd state is connected only with odd states, and each even state is connected only with even states, we have a reducible Markov chain. Hence, we are allowed to split the initial chain into two separate ones, which are more simple to analyze. Thus, we end up with two non-reducible Markov chains, which we analyze both in the same way as follows.

Using Markov theory, we calculate the state probabilities using the set of equations for the equilibrium probability p_k.

At first, we examine the chain with the even states as depicted in figure 4.10. Since the maximum distance D of two nodes is $2 * L$, the last state will always be an even number, so D must be included in this chain. Furthermore, let us denote every transition from state i towards larger distance as r_i, and every transition from state i towards smaller distance as s_i. The probability of two nodes being in distance i, which is interpreted as being in state i, is denoted as P_i. Accordingly, let

$$P_i = \lim_{t\to\infty} P_k(t) \tag{4.2}$$

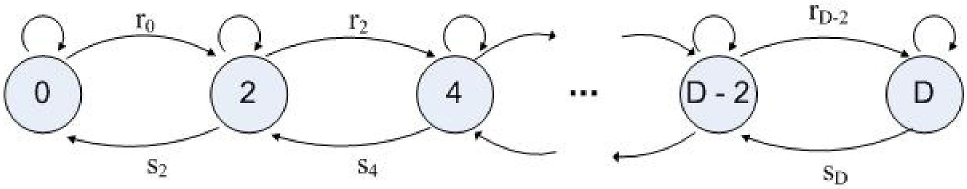


Figure 4.10: Even subset of states for taxicab distance

Then, we have the following system of equations that apply to the chain with the even states

$$r_0 \cdot P_0 = s_2 \cdot P_2 \tag{4.3}$$

$$r_{2j} \cdot P_{2j} = s_{2j+2} \cdot P_{2j+2} \tag{4.4}$$

for j$\geq$ 1

From the above two equations, we observe that it is easy to write every state probability with respect to the first state probability P_0. Thus, we have

$$P_{2j+2} = r_{2j} \cdot P_{2j}/s_{2j+2} \tag{4.5}$$

According to the flow conservation relation, there is one more requirement, that the sum of all the state probabilities must be equal to one. Hence,

$$\sum_{k=0}^{D/2} P_{2k} = 1 \tag{4.6}$$

$$\begin{aligned}
1 &= P_0 + P_2 + ... + P_D \\
&= P_0 + P_0 \cdot \frac{r_0}{s_2} + P_0 \cdot \frac{r_0}{s_2} \cdot \frac{r_2}{s_4} + ... + P_0 \cdot \frac{r_0}{s_2} \cdot \frac{r_2}{s_4} ... \cdot \frac{r_{D-2}}{s_D} \\
&= P_0 \cdot [1 + \frac{r_0}{s_2} + \frac{r_0}{s_2} \cdot \frac{r_2}{s_4} + ... + \frac{r_0}{s_2} \cdot \frac{r_2}{s_4} ... \cdot \frac{r_{D-2}}{s_D}] \\
&= P_0 \cdot [1 + \prod_{k=0}^{(D-2)/2} [\frac{r_{2k}}{s_{2k+2}}]
\end{aligned} \tag{4.7}$$

$$\begin{aligned} P_0 &= \frac{1}{[1 + \frac{r_0}{s_2} + \frac{r_0}{s_2} \cdot \frac{r_2}{s_4} + ... + \frac{r_0}{s_2} \cdot \frac{r_2}{s_4} \cdot ... \cdot \frac{r_{D-2}}{s_D}]} \\ &= P_0 \cdot [1 + \prod_{k=0}^{(D-2)/2} [\frac{r_{2k}}{s_{2k+2}}] \end{aligned} \tag{4.8}$$

Consequently, for the general state probability we obtain

$$P_{2k} = P_0 \cdot \prod_{k=0}^{(D-2)/2} [\frac{r_{2k}}{s_{2k+2}}] \tag{4.9}$$

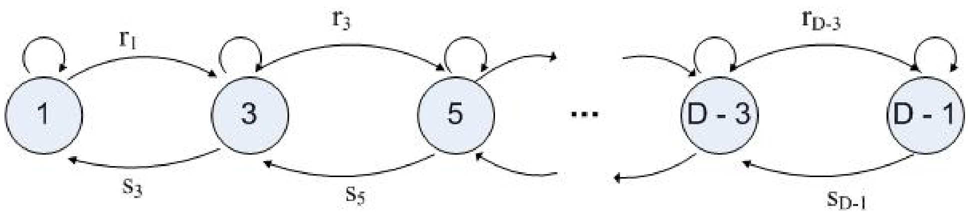


Figure 4.11: Odd subset of states for taxicab distance

Accordingly, if we apply the same set of equations to the chain with the odd states, as shown in figure 4.11 we will get for j$\geq$ 0

$$r_1 \cdot P_1 = s_3 \cdot P_3 \tag{4.10}$$

$$r_{2j+1} \cdot P_{2j+1} = s_{2j+3} \cdot P_{2j+3} \tag{4.11}$$

According to the flow conservation relation, there is one more requirement, that the sum of all the state probabilities must be equal to one. Hence,

$$\sum_{k=0}^{D-1} P_{2k+1} = 1 \tag{4.12}$$

$$\begin{aligned} 1 &= P_1 + P_3 + ... + P_{D-1} \\ &= P_1 + P_1 \cdot \frac{r_1}{s_1} + P_1 \cdot \frac{r_1}{s_3} \cdot \frac{r_3}{s_5} + ... + P_1 \cdot \frac{r_1}{s_3} \cdot \frac{r_3}{s_5} ... \cdot \frac{r_{D-3}}{s_{D-1}} \\ &= P_1 \cdot [1 + \frac{r_1}{s_3} + \frac{r_1}{s_3} \cdot \frac{r_3}{s_5} + ... + \frac{r_1}{s_3} \cdot \frac{r_3}{s_5} ... \cdot \frac{r_{D-3}}{s_{D-1}}] \\ &= P_1 \cdot [1 + \prod_{k=0}^{(D-4)/2} [\frac{r_{2k+1}}{s_{2k+3}}] \end{aligned} \tag{4.13}$$

$$\begin{aligned} P_1 &= \frac{1}{[1 + \frac{r_1}{s_3} + \frac{r_1}{s_3} \cdot \frac{r_3}{s_5} + ... + \frac{r_1}{s_3} \cdot \frac{r_3}{s_5} \cdot ... \cdot \frac{r_{D-3}}{s_{D-1}}]} \\ &= P_1 \cdot [1 + \prod_{k=0}^{(D-4)/2} [\frac{r_{2k+1}}{s_{2k+3}}] \end{aligned} \tag{4.14}$$

Consequently, for the general state probability we obtain

$$P_{2k+1} = P_1 \cdot \prod_{k=0}^{(D-4)/2} [\frac{r_{2k+1}}{s_{2k+3}}] \tag{4.15}$$

4.5.2 Transition Probabilities

The transition probabilities between each state, are easy to calculate if we consider that each node has *4* possible directions to move. Since we examine how a pair of nodes moves each time, we have *16* different combinations of motions. Some of these combinations lead to the same taxicab distance between the two nodes. If the initial distance of the nodes is more than *1*, there can be multiple positions on grid yielding the same distance, and each one affects the results of the combinations in a different way. For example two nodes in initial taxicab distance *D=2*, can be collinear, or not. If we examine the two cases separately, we will see that the probabilities of going to the states *0* and *4* are different. After calculating all the possible positions of the pair of nodes on the grid that yield the same taxicab distance, and the relevant combinations of motions, the transition matrix is filled as follows for a network consisted of N nodes in a square grid of length L:

The table below shows the transition probabilities in case that the two nodes are collinear.

$$\begin{pmatrix}
4/16 & 0 & 12/16 & 0 & 0 & ... & 0 & 0 & 0 & 0 \\
0 & 9/16 & 0 & 7/16 & 0 & ... & 0 & 0 & 0 & 0 \\
1/16 & 0 & 8/16 & 0 & 7/16 & ... & 0 & 0 & 0 & 0 \\
0 & 1/16 & 0 & 8/16 & 0 & 7/16 & 0 & ... & 0 & 0 \\
0 & 0 & 1/16 & 0 & 8/16 & 0 & 7/16 & 0 & ... & 0 \\
... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\
0 & 0 & 0 & ... & 1/16 & 0 & 8/16 & 0 & 7/16 & 0 \\
0 & 0 & ... & 1/16 & 0 & 8/16 & 0 & 7/16 & 0 & 0 \\
0 & 0 & 0 & ... & 1/16 & 0 & 8/16 & 0 & 7/16 & 0 \\
0 & 0 & ... & 0 & 0 & 0 & 0 & 12/16 & 0 & 4/16
\end{pmatrix}$$

If the two nodes are not collinear, but their coordinates differ by one, either this is true for Dx, or Dy or for both of them, then we have another transition matrix, which is the product of the two following tables. The first one provides the number of different initial position of nodes that fulfill the aforementioned condition, and the latter provides the probabilities calculated the same way as before.

$$\begin{pmatrix}
0 \\ 0 \\ 1 \\ 2 \\ 2 \\ ... \\ 2 \\ 0 \\ 0 \\ 0
\end{pmatrix}$$

Here we need to note that if the two nodes are initially in the same position, or at distance *1*, they are considered to be collinear, and thus this case is included in the

previous transition matrix. Each line of the table represents their initial distance. As long as this distance is smaller than the dimension of the grid, then there are only two ways that the condition *Dx=1* or *Dy=1* can be met. Either
Dx=l-1
Dy=1 or
Dx=1
Dy=l-1

If the distance between the pair of nodes is larger than the dimension of the grid plus one (*D>L+1*), then this condition can no longer be met, so the table is filled with zeros.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2/16 & 0 & 10/16 & 0 & 4/16 & 0 & ... & 0 & 0 & 0 \\ 0 & 3/16 & 0 & 9/16 & 0 & 4/16 & 0 & ... & 0 & 0 \\ 0 & 0 & 3/16 & 0 & 9/16 & 0 & 4/16 & 0 & ... & 0 \\ 0 & 0 & 0 & 3/16 & 0 & 9/16 & 0 & 4/16 & ... & 0 \\ 0 & 0 & 0 & 0 & 3/16 & 0 & 9/16 & ... & 0 & 0 \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & 3/16 & 0 & 9/16 & 0 & 4/16 & 0 \\ 0 & 0 & 0 & ... & 0 & 3/16 & 0 & 9/16 & 0 & 4/16 \end{pmatrix}$$

It is clear that in the above transition matrix, there can be only three non zero values per line, as the distance can either be increased or decreased by two, or stay the same.

The following two tables, examine the remaining cases, for the initial positions of the nodes on the grid. The first of the following tables, represents the number of different initial positions of nodes on the grid for this specific case. As long as the initial taxicab distance D is less than the dimension L of the grid, then there are D-3 different positions that all yield the same transition probabilities described in the latter table. For the cases that the distance is 6 this rule does not hold. If the distance is larger than the grid dimension, then the number of combinations is given by the formula *2L-D*, where D is the corresponding line in the table.

$$\begin{pmatrix} 0 & ... & 0 \\ 0 & ... & 0 \\ 0 & ... & 0 \\ 0 & ... & 0 \\ 1 & ... & 1 \\ 2 & ... & 2 \\ 5 & ... & 5 \\ 4 & ... & 4 \\ 5 & ... & 5 \\ l-3 & ... & l-3 \\ ... & ... & ... \\ L-3 & ... & L-3 \\ 2L-l & ... & 2L-l \\ ... & ... & ... \\ 2L-l & ... & 2L-l \end{pmatrix}$$

The following table includes the transition probabilities for the case described above.

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 4/16 & 0 & 8/16 & 0 & 4/16 & 0 & ... & 0 \\
0 & 0 & 0 & 4/16 & 0 & 8/16 & 0 & 4/16 & ... & 0 \\
0 & 0 & 0 & 0 & 4/16 & 0 & 8/16 & ... & 0 & 0 \\
... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\
0 & 0 & 0 & ... & 4/16 & 0 & 8/16 & 0 & 4/16 & 0 \\
0 & 0 & 0 & ... & 0 & 4/16 & 0 & 8/16 & 0 & 4/16
\end{pmatrix}$$

In order to take into consideration all the different cases, we suggest to take a weight average of the transition probabilities, using as weight, the number of combinations for each case.

4.6 Hop count Analysis

In this project, we observe the hop count between different pairs of nodes, and how this metric evolves over time. When two nodes not directly connected to each other want to communicate, some intermediate nodes will forward the packets to the destination. The number of times that the packets will be transmitted until they are received by the destination, is equal to the hop count. Since every node moves, this number will change between consecutive time slots whenever nodes move outside the range of their former neighbors. The value of hop count in the next time slot, can be predicted with a certain probability, with knowledge only of the current value of it.

$$Pr[H(t+1)|H(t), H(t-1), ..., H(1)] = Pr[H(t+1)|H(t)] \tag{4.16}$$

In the two-dimensional random walk mobility model each node moves as a random walker on a two-dimensional lattice. The time is discrete and at each time step each node has a probability of 1/4 of hopping to a position above, below, left or right of its current position. If the node is positioned on a boundary, then instead of hopping off the lattice it is reflected towards the opposite direction. The stationary distribution of the location of a two-dimensional random walk on a square lattice is uniform over the area. This property is a consequence of the fact that a two dimensional random walk can be constructed from two independent one-dimensional random walk, and the stationary location of a symmetric random walk in one-dimension is uniform.

Using Markov theory to analyze the way that this metric changes between the possible values is very complicated. The Markov chain in figure 4.12 represents all the states of the hop count and as we can see, every odd state is connected to every other odd state, and the same holds for even states. In contrast to the taxicab analysis, here the chain cannot be reduced or separated, as the first state is part of both odd and even subsets of the chain.

Thus, our analysis is continued in a different way. At first, we proceed as [1] with the difference that nodes move on a grid. Lets assume two nodes are 2-hop neighbors, this means that they don't have a direct link between each other but can communicate through at least one intermediate node. Therefore at least one intermediate node must be at the marked area (the circles define the coverage area of each node) in figure 4.13 to preserve connectivity between nodes s, d. In our topology, nodes are placed only on vertices of a grid, and are uniformly distributed during all time slots since they perform random walk. Hence, instead of an area A as shown in the figure 4.13, we have a number of vertices where at least one node must be. In this section, we are interested in estimating the distribution of the hop count in the future time slots, given the current hop count, or the current taxicab distance. If we know the current distance of the two nodes, then, from the taxicab analysis presented in previous section we can estimate the future taxicab distance of this pair (Prtaxicab(t+1)|taxicab(t)). Consequently, we know a lower limit of how many intermediate nodes are needed to keep the two nodes connected (combinatory problem). In a grid of area $L * L$, and N nodes, assuming that we can replace each node back to the set after using it, the following formula gives us the probability of having one node at the exact vertex that we need it to be:

$$Pr[i] = \begin{pmatrix} N \\ 1 \end{pmatrix} \cdot \left(\frac{1}{L * L}\right)^{H(t+1)-1} \tag{4.17}$$

The steps we described so far, are summarized below for two nodes A and B:

1. We get informed about the current distance of A, B
2. We estimate their distance at time t+1, through the distribution of Pr(taxicab(t+1)|taxicab(t))

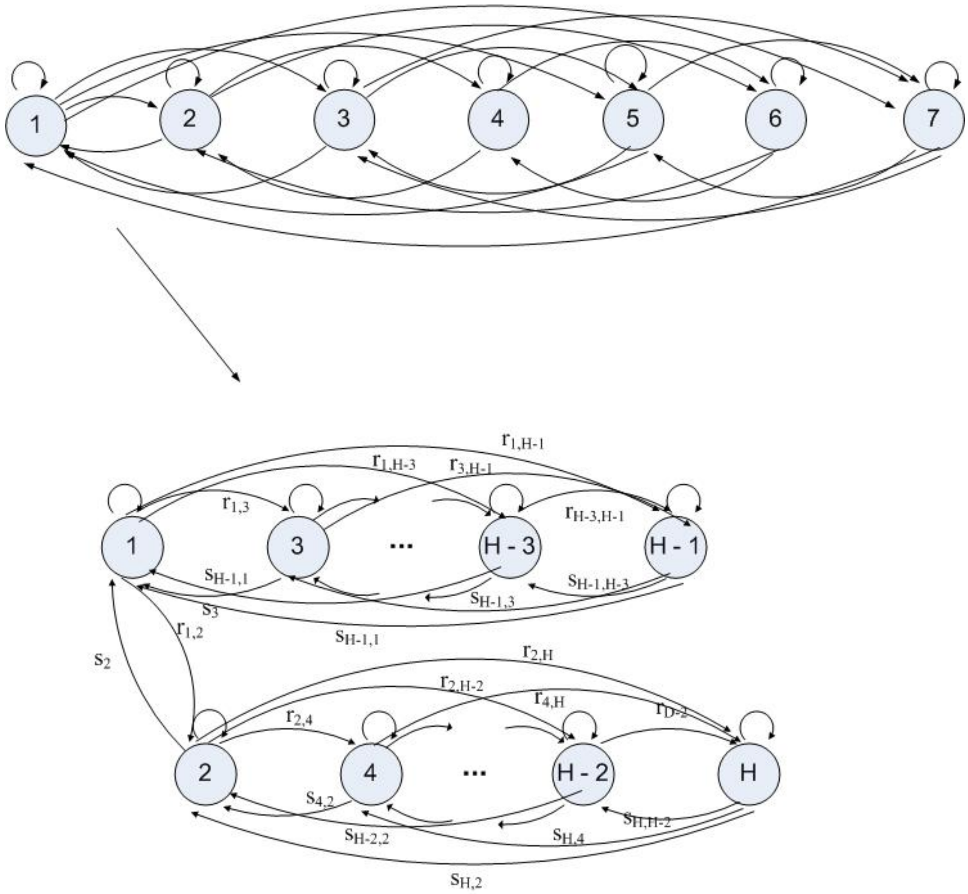


Figure 4.12: Markov chain for hop count metric

3. We estimate the probability Pr(hop(t+1)|taxicab(t+1))
4. We calculate the probability that each node of the new route will be at the correct place with replacement

To be more specific, in case of short length paths, we can calculate the number of all possible paths that exist between a pair of nodes, and then find the probability that a node can be at each intermediate point so to have a connected path. Thus, if we denote the hop count at the next time slot as H', and the taxicab distance at the next time slot as taxicab', then for paths of different length we can argue the following:

- Given that taxicab'=0, P[H'=1]=1 deterministically for reasons that we have mentioned in previous chapters.
- Given that taxicab'=1, Pr[H'=1]=1, because the range of each node is equal to one grid unit, so nodes can communicate directly.
- Given that taxicab'=2, we have

$$Pr[H' = 2] = \binom{N}{1} \cdot (\frac{1}{L * L})^1 \tag{4.18}$$

or

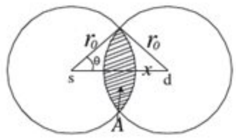


Figure 4.13: Two hop neighbors separated by a known distance x as depicted in [1]

$$Pr[H' = 4] = \begin{pmatrix} N \\ 1 \end{pmatrix} \cdot (\frac{1}{L * L})^3 \quad (4.19)$$

or

$$Pr[H' = 6] = \begin{pmatrix} N \\ 1 \end{pmatrix} \cdot (\frac{1}{L * L})^5 \quad (4.20)$$

because as we already explained the nodes that are at even taxicab distance, will have an even number of hop count as well.

- Given that taxicab'=3, we have

$$Pr[H' = 3] = \begin{pmatrix} N \\ 1 \end{pmatrix} \cdot (\frac{1}{L * L})^2 \quad (4.21)$$

or

$$Pr[H' = 5] = \begin{pmatrix} N \\ 1 \end{pmatrix} \cdot (\frac{1}{L * L})^4 \quad (4.22)$$

At this point we note that we don't count all the possible paths between two nodes, but instead, we stop at a path of 6 hops. Furthermore, we have observed that if the nodes are collinear, then some routes are not possible. However, if the grid is sufficiently large, the ratio of collinear nodes to non collinear, tends to zero, which leads to the conclusion that we can take ignore the collinear case.

This analysis can also be used for cellular networks, [4], with some variations. In such networks, where nodes are in a hexagonal area, each node has 6 neighbors, and six equiprobable options for its next position (1/6 probability for each option). Moreover,

$$d(t+1) = \begin{cases} d(t) + 2 \\ d(t) + 1 \\ d(t) \\ d(t) - 1 \\ d(t) - 2 \end{cases}$$

Although such cellular networks have many common properties with the mobile networks that we analyze in this thesis, further and more explicit analysis of such networks is left for future work.

Chapter 5

Conclusion

In this chapter we conclude the project thesis and state the future work that remains after our contribution.

5.1 Conclusion

Throughout this project we used a grid topology where nodes could perform random walk on the vertices of it to investigate the two forwarding schemes. This decision made the analysis feasible and the simulation results easier to understand. In order to come to compare the performance of source forwarding versus that of intermediate forwarding we use two metrics; the taxicab distance and the hop count metric.

We present four different distributions, that stem from four conditional probabilities which we analyze in chapter 3.6 and explain them in the relevant figures. The results show that there is a characteristic value for all initial states (either the state refers to hop count value or to taxicab distance) and both longer and shorter distances tend to this value.

Intermediate forwarding seems very promising as it performs better in terms of saving resources and time, but in order to deploy it, we need to have control of the routers of each network. This could be one of the reasons justifying why it is not applied yet.

5.2 Future Work

Some topics relevant with this project remain open for further investigation. As such we mention the analytical work which supports our simulations. It is very difficult to build a model for this kind of networks, even in the most simple case of grid. We suggest an approach using Markov theory for grid topology. Such an approach can be used for continuous time and space variables, but we leave this open for future work.

Furthermore, nodes in our model perform random walk. In a more realistic scenario nodes could follow a different routing protocol according to a social network.

Chapter 6

Appendix

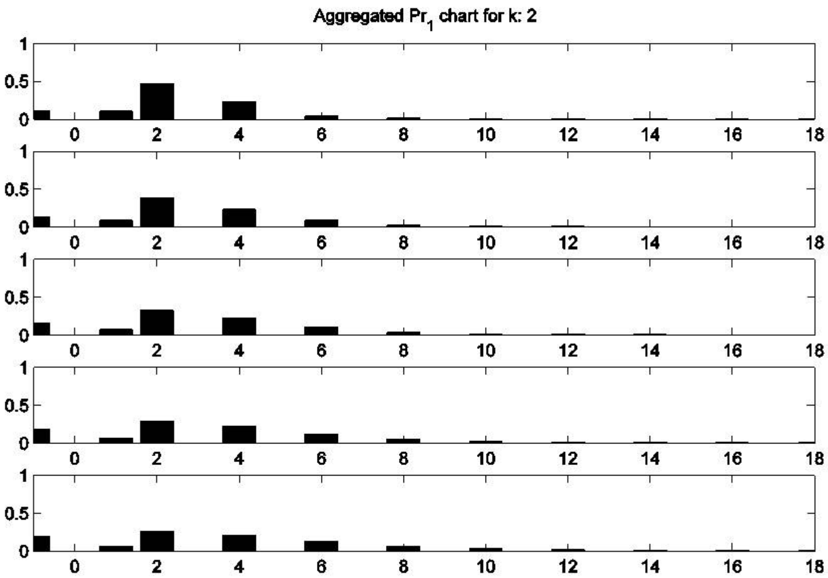


Figure 6.1: Hop count distribution, initial hop count 2

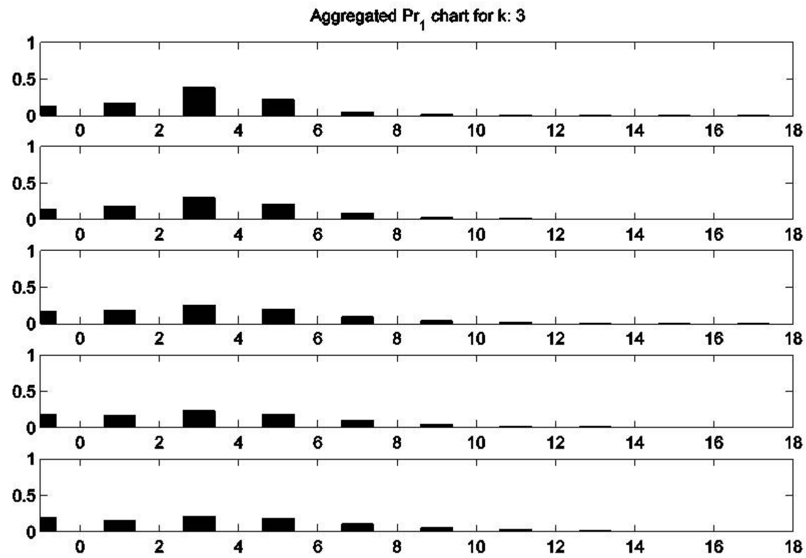


Figure 6.2: Hop count distribution, initial hop count 3

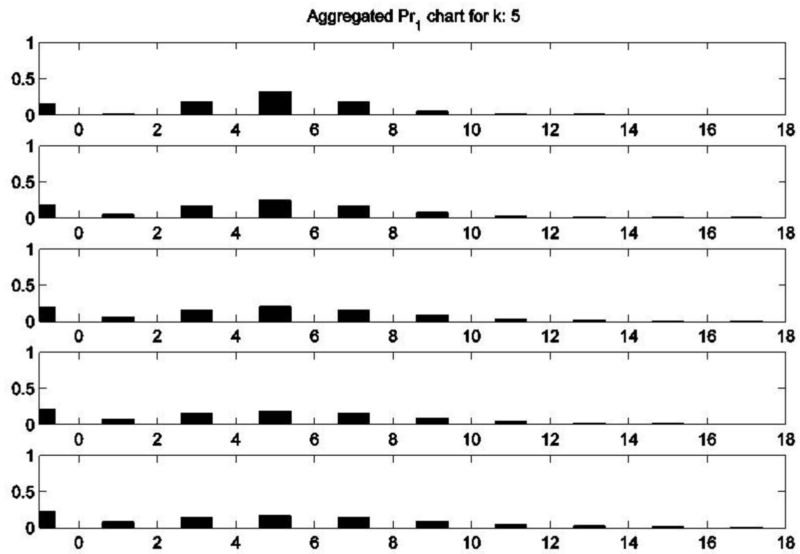


Figure 6.3: Hop count distribution, initial hop count 5

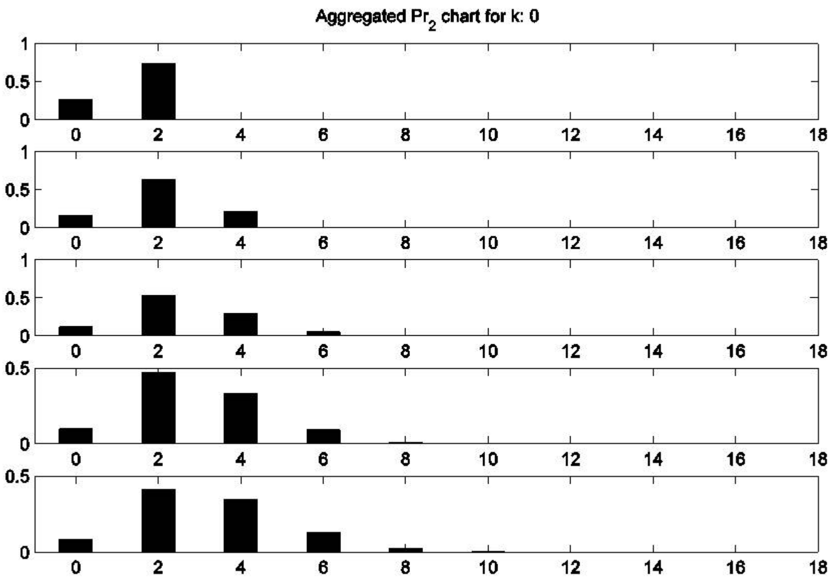


Figure 6.4: Taxicab distribution, initial taxicab distance 0

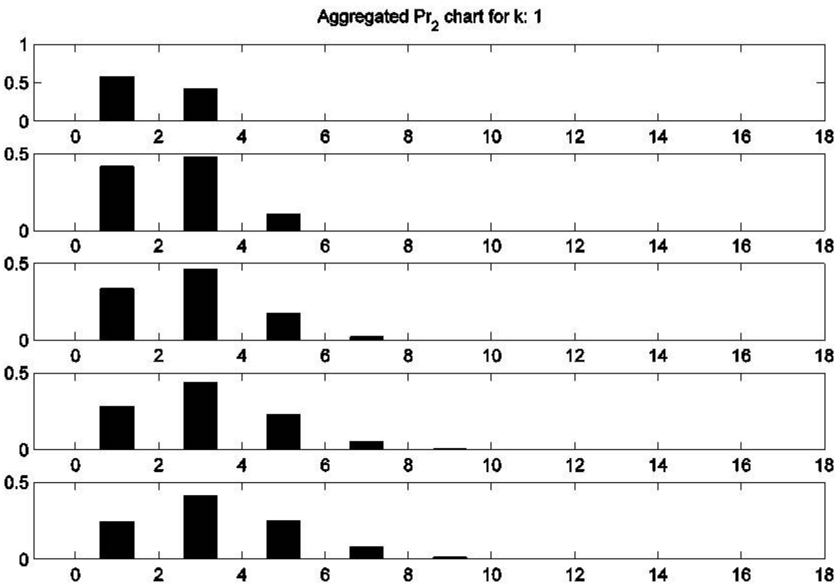


Figure 6.5: Taxicab distribution, initial taxicab distance 1

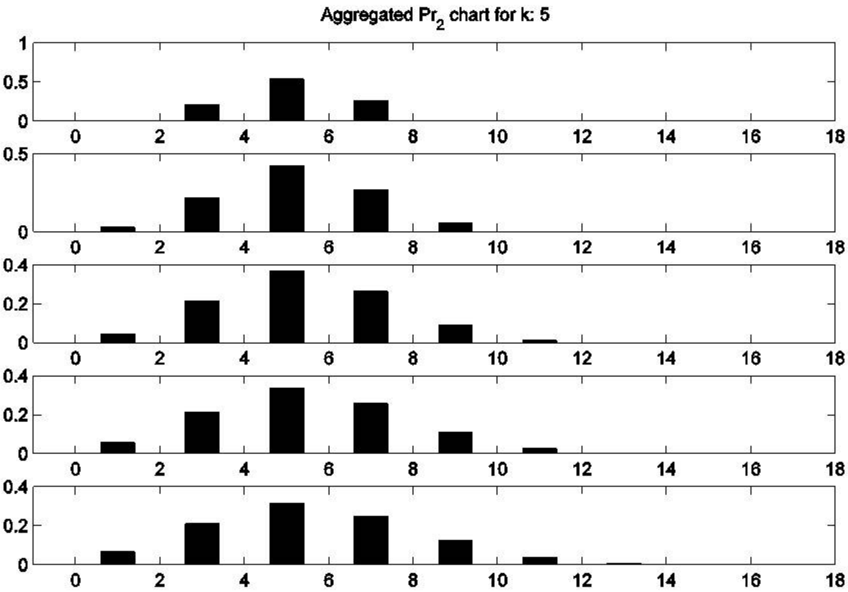


Figure 6.6: Hop count distribution, initial taxicab distance 5

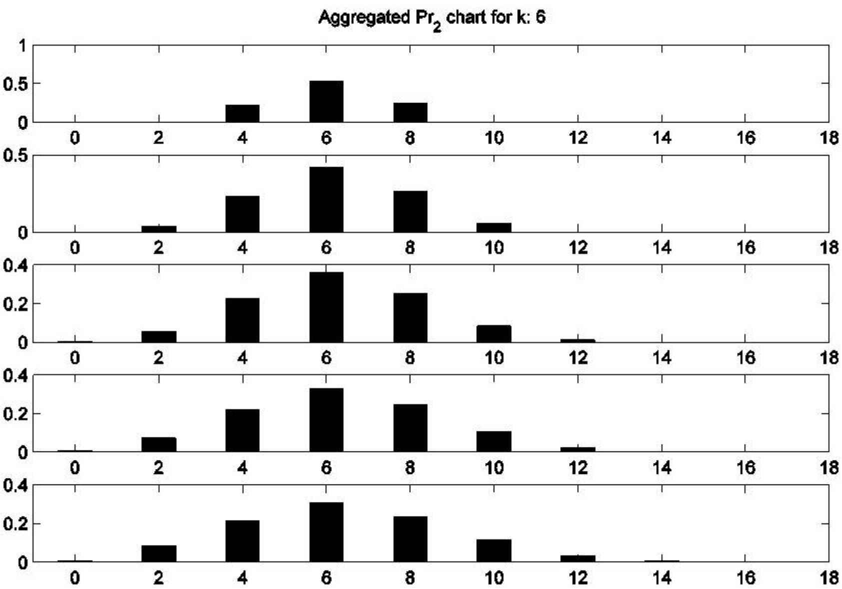


Figure 6.7: Hop count distribution, initial taxicab distance 6

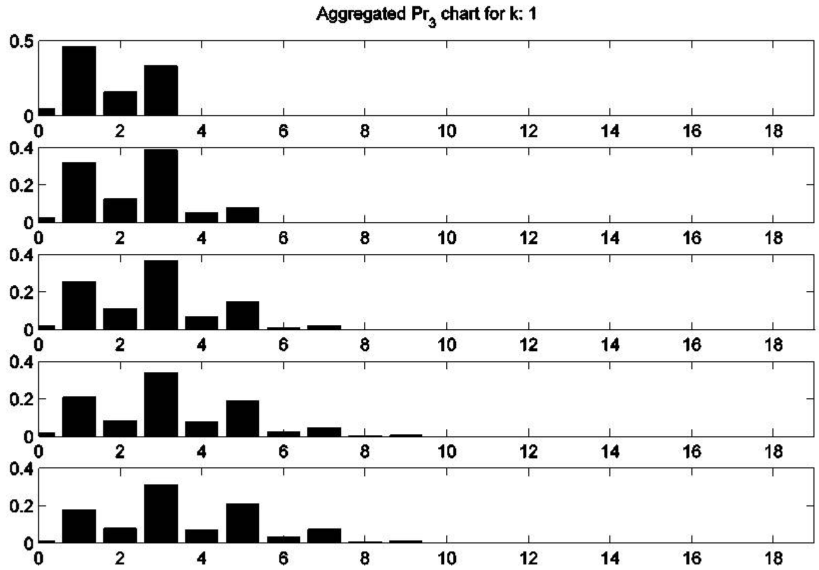


Figure 6.8: Taxicab distribution, initial hop count 1

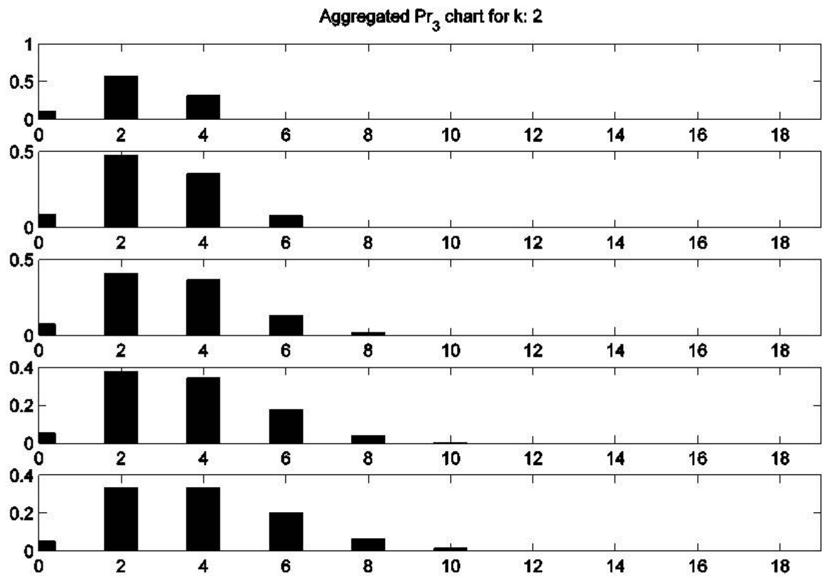


Figure 6.9: Taxicab distribution, initial hop count 2

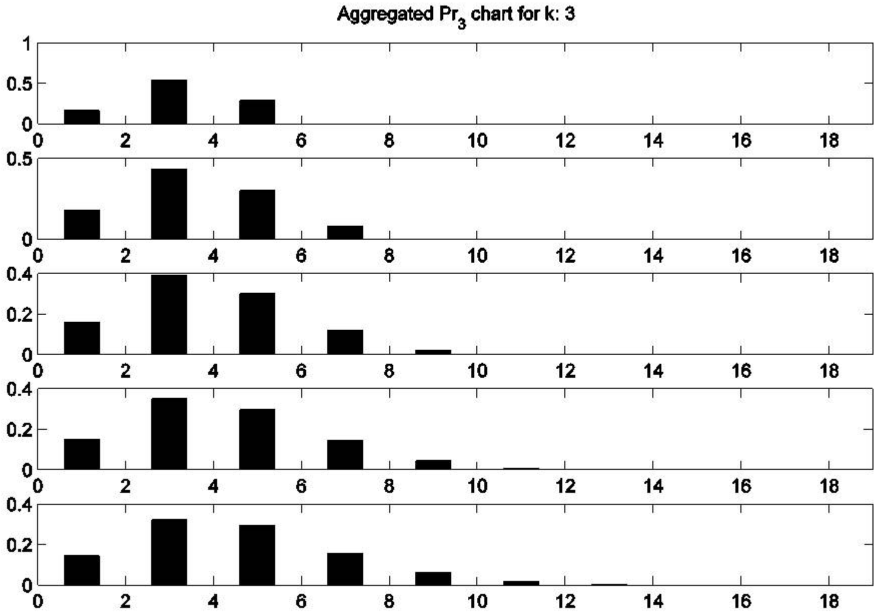


Figure 6.10: Taxicab distribution, initial hop count 3

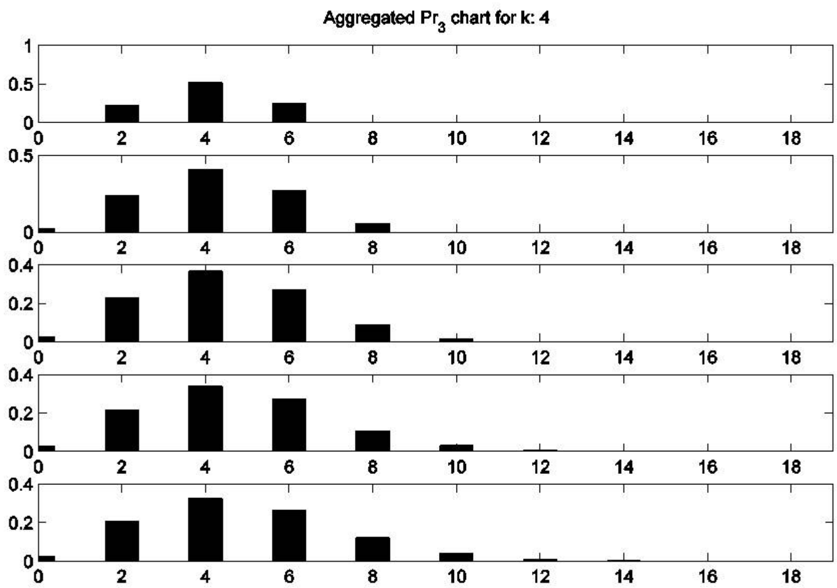


Figure 6.11: Taxicab distribution, initial hop count 4

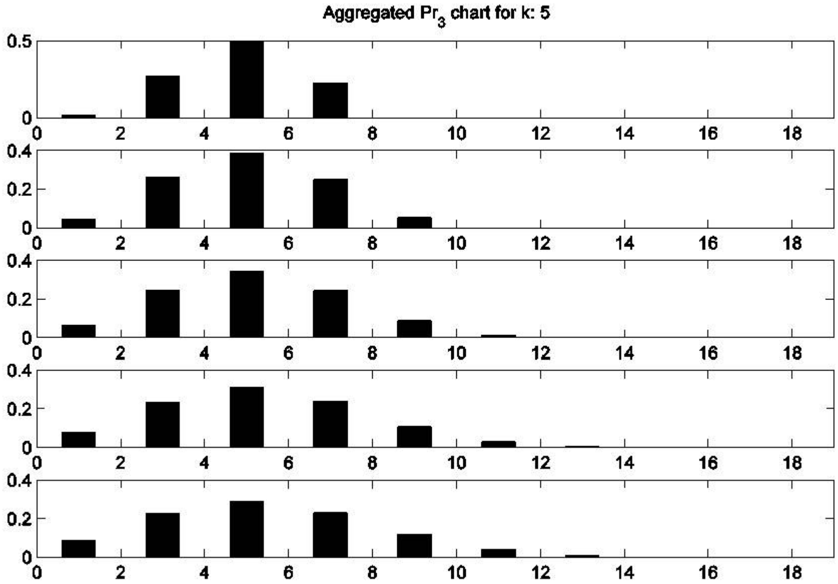


Figure 6.12: Taxicab distribution, initial hop count 5

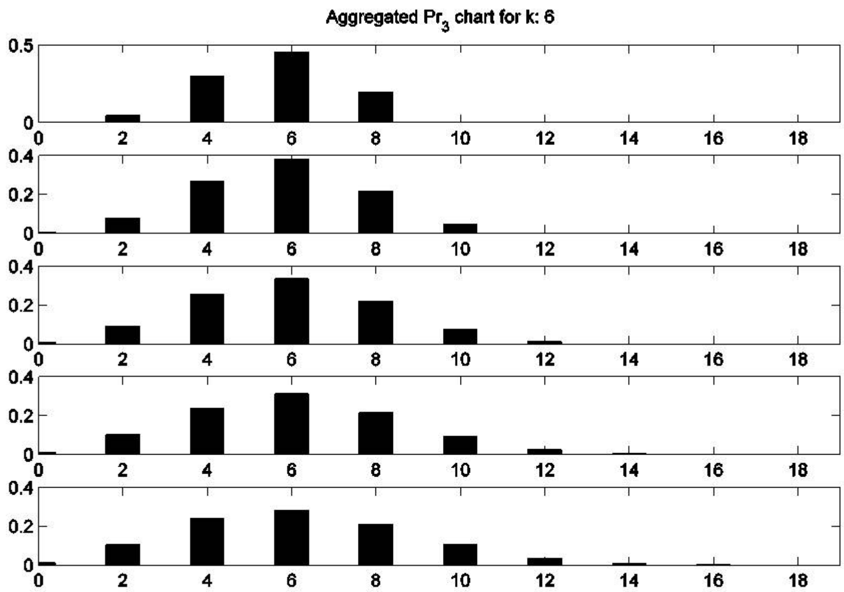


Figure 6.13: Taxicab distribution, initial hop count 6

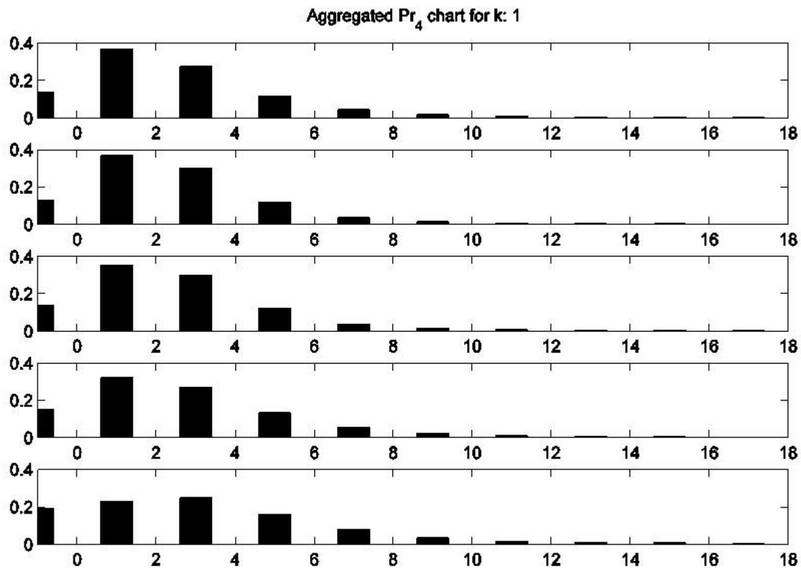


Figure 6.14: Hop count distribution, initial taxicab distance 1

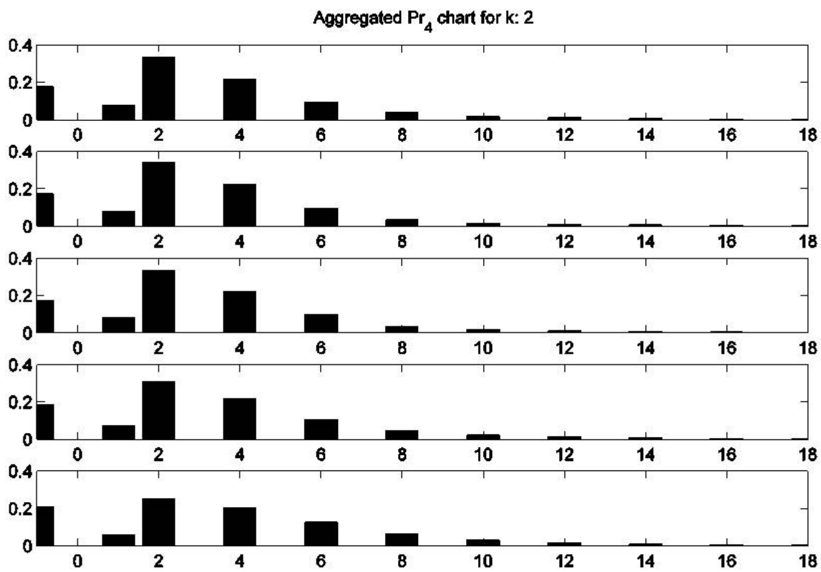


Figure 6.15: Hop count distribution, initial taxicab distance 2

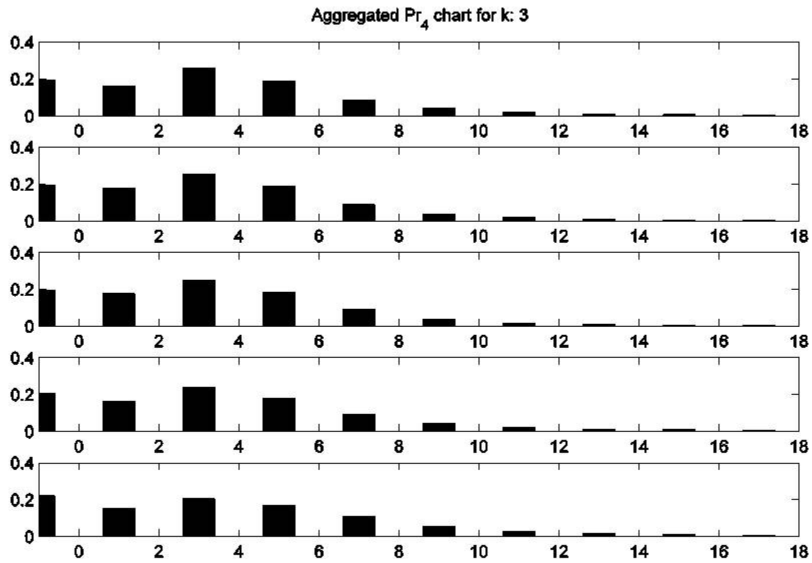


Figure 6.16: Hop count distribution, initial taxicab distance 3

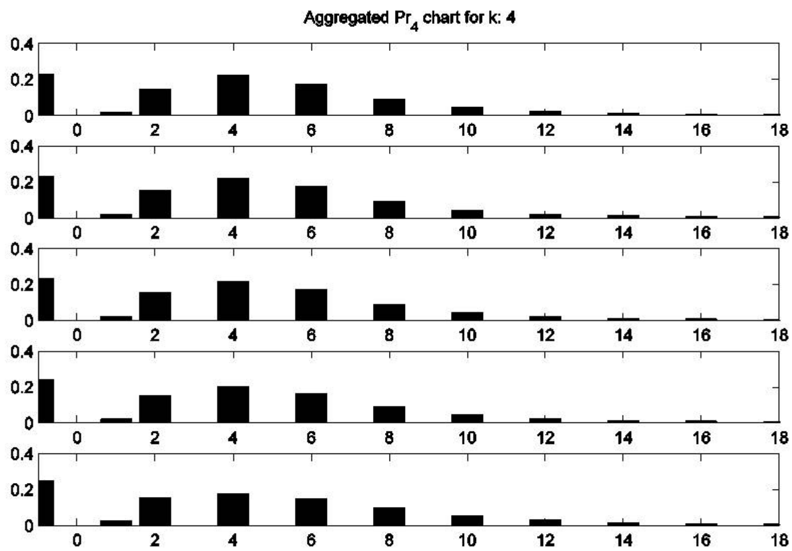


Figure 6.17: Hop count distribution, initial taxicab distance 4

References

[1] Xiaoyuan Ta, Guoqiang Mao, Brian D.O. Anderson, "Evaluation of the Probability of k-hop Connection in Homogenous Wireless Sensor Networks," *IEEE GLOBECOM 2007 Proceedings.*

[2] Simon Heimlicher, Merkouris Karaliopoulos, Hanoch Levy and Thrasyvoulos Spyropoulos, "On Leveraging Partial Paths in Partially-Connected Networks," *IEEE INFOCOM 2009.*

[3] Simon Heimlicher, Martin May and Bernhard Plattner, "The Transport Layer Revisited," *2nd IEEE International Conference on Communication System Software and Middleware, IEEE COMSWARE 2007.*

[4] Bernd Meyer, "On Route Lifetime in Multihop Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 2, pp. 366–376, 2003.

[5] Bernd Meyer, "On the hop count statistics for randomly deployed wireless networks," *Int. J. Sensor Networks*, vol. 1, pp. 89–102, 2006.

[6] Stefen C. Peterson and Peter B. Noble, "A Two Dimensional Random Walk Analysis of Human Granulocyte Movement," *Biophysical*, vol. 12, pp. 1048–1055, 1972.

[7] Ming Zhong, Kai Shen, Joel Seiferas, "The Convergence-Guaranteed Random Walk and Its Applications in Peer-to-Peer Networks," *IEEE Transactions on Cimputers*, vol. 57, pp. 619–633, 2008.

[8] Vangular S. Alagar, "The Distribution of the Distance between Random Points," *Applied Probability*, vol. 13, pp. 558–566, 1976.

[9] Simon Heimlicher, Merkouris Karaliopoulos, Hanoch Levyn and Martin May, "End-to-End vs. Hop-by-Hop Transport under Intermittent Connectivity," *ICST Autonomics 2007*, vol. 302.

[10] Robin Groevelt, Philippe Nain, Ger Koole, "The message delay in mobile ad hoc networks," *Performance Evaluation*, vol. 62, pp. 210–228, 2005.

[11] Christopher Z. Mooney, *Monte Carlo Simulation.* Sage Publications, first ed., 1997.

[12] " Introduction to Monte Carlo Methods." http://www.phy.ornl.gov/csep/mc/node1.html.

[13] Anastasios Papoulis and S. Unnikrishna Pillai, *Probability, Random Variables and Stochastic Processes.* McGraw-Hill, fourth ed., 2002.