

Department of Mathematics and Computer Science
Bar Ilan University

Routing in Gossip Networks

Roni Parshani

Submitted in partial fulfillment of the requirements for the Master's degree in the
Department of Mathematics and Computer Science, Bar Ilan University

Abstract

The purpose of this work is to investigate the "Gossip Network" model. In this model, agents traveling in a dynamic network locally exchange information using ad hoc communication. In dynamic networks the edges building the network contain weights that vary in time. Therefore one of the important issues in dynamic networks is how the information about the network state is distributed. The gossip network model introduces a method where the agents traveling in the network also distribute the information regarding the network state.

Although the gossip network model is adequate for many real time problems, we focus on an implementation of the gossip network model in the field of transportation networks.

Because of the complexity of mathematically analyzing dynamic networks the investigation was done via simulation. For this purpose a special micro simulation tool was built in order to support the use of gossiping between individual cars.

The first goal of this work is to analyze the characteristics of the information expansion in the gossip network. For this purpose we have defined a series of measurements: the expansion rate of the information in the network, the probability of learning information on every edge and the average delay in receiving the information. Using these measurements we evaluate the quality of the information expansion and show how the gossip network properties affect the quality of information expansion.

The second goal is to investigate the influence of the gossiping model on the routing guidance problem. This is achieved by comparing the same routing algorithm for different networks. The different networks are all dynamic networks but each have a different information layer. This way the gossiping model is compared to other information layer models like the centralized and decentralized models.

We present some surprising results regarding the conditions in which the gossiping model is optimal. We also show results regarding routing capabilities in a gossiping model compared to that of the centralized information model.

Acknowledgments

I wish to thank all those who assisted me in various ways during the period it took to complete this thesis.

First, I wish to thank my supervisor Prof. Sarit Krauss and my instructor Dr. Yuval Shavitt for their guidance and support during the whole period – from the stage of raising ideas until the stage of reading and re-reading the drafts.

My thanks to Karen Reiffman for her exceptional editorial assistance which were given in a very short notice.

Finally, I would like to thank my wife for encouraging me and for taking care of our children so I could quietly complete my work.

Contents

1. Introduction.....	6
2. Related Work.....	9
2.1 Simulations of Transportation Networks.....	9
2.2 Routing in Transportation Networks.....	10
2.3 Gossiping.....	11
3. Gossip Networks.....	12
3.1 Gathering Information.....	12
3.2 Exchanging Information.....	12
3.2.1 Gossip car percent.....	13
3.2.2 Information exchange.....	13
3.3 Information Expansion.....	13
3.3.1 EIATBU measurement.....	14
3.3.2 EID measurement.....	14
3.4 Exploring characteristics of Gossip networks.....	14
4. Routing Guidance in Dynamic Transportation Networks.....	15
4.1 Routing Guidance Comparison.....	15
4.2 Types of Information Layers.....	16
4.3 Routing Algorithm	17
4.4 Routing Performance.....	18

4.5 Special cars percentage.....	18
5. Simulation.....	19
5.1 Simulation structure.....	19
5.2 Simulation model.....	19
5.3 Physical Network model.....	20
5.4 Movement Model.....	21
5.5 Traffic model.....	21
5.6 Simulation Configuration	22
5.7 Simulation Modes.....	23
5.8 Simulation Stages.....	23
6. Tests & Results.....	25
6.1 Gossip network characteristics.....	25
6.2 Routing guidance in Gossip networks.....	35
6.3 Partial information.....	42
7. Conclusions.....	45
8. Future work.....	47
9. References	49

1. Introduction

The purpose of this work is to investigate the "Gossip Network" model [35]. In this model, agents traveling in a dynamic network [38-39] locally exchange information using ad hoc communication. In this paper the act of locally exchanging information is named "gossiping". In dynamic networks the edges building the network contain weights that vary in time. Therefore one of the important issues in dynamic networks is how the information about the network state is distributed. The gossip network model introduces a method where the agents traveling in the network also distribute the information regarding the network state. This method of distribution has two important characteristics: first, there is no need for any additional infrastructure to support the distribution of the information. Second, there is a mutual influence between the information expansion and the routing capabilities of the agents. Because the agents traveling in the network are responsible for distributing the information, the way they move influences the way the information is spread throughout the network. However, the information about the network state itself is also an important factor in the routing algorithm.

The gossip network model is actually a combination of two network layers. The first, the physical Network layer, is compatible with the dynamic network model. The second, the information layer, is compatible with the gossiping model.

Although the gossip network model is adequate for many real time problems, we focus on an implementation of the gossip network model in the field of transportation networks [25-30]. In gossip transportation networks, the physical network is composed of roads and junctions. Each road contains a weight that is equivalent to the time it takes to traverse the road. A road's weight varies according to the load of traffic and as a result of unpredictable incidents like accidents. The information exchanged by gossiping includes road conditions, traffic jams and other data regarding the network state. The exchange of information using gossiping is done in

one of two ways: between cars passing one another or between cars while waiting at junctions.

In order to support gossiping, a special device needs to be installed in every car participating in the gossiping process.

Because of the complexity of mathematically analyzing dynamic networks the investigation was done via simulation. For this purpose a special micro simulation tool [6-16] was built in order to support the use of gossiping between individual cars.

The first goal of this work is to analyze the characteristics of the information expansion in the gossip network. For this purpose we have defined a series of measurements: the expansion rate of the information in the network, the probability of learning information on every edge and the average delay in receiving the information. Using these measurements we evaluate the quality of the information expansion and show how the gossip network properties affect the quality of information expansion. For example, one of the interesting properties of the gossip network is the percentage of cars participating in the gossiping process. Since the act of gossiping requires special equipment it is reasonable to assume that not all of the cars in the network will be able to perform the act of gossiping. The "gossip car percent" has a major effect on the quality of information expansion.

The second goal is to investigate the influence of the gossiping model on the routing guidance problem [21-30]. This is achieved by comparing the same routing algorithm for different networks. The different networks are all dynamic networks but each have a different information layer. This way the gossiping model is compared to other information layer models like the centralized and decentralized models.

We present some surprising results regarding the conditions in which the gossiping model is optimal. We also show results regarding routing capabilities in a gossiping model compared to that of the centralized information model.

The rest of this thesis is divided as follows:

Chapter two, "Related work" presents other similar work done on the subject and compares between them and our work. Chapter three, "Gossip Networks" analyzes the characteristics of gossip networks. Chapter four, "Routing Guidance in Dynamic Transportation Networks" compares between the gossip model and other information layers. Chapter five, "Simulation" describes the simulation model used and which assumptions we made. It also describes the simulation tool built. Chapter six, "Tests

& Results" presents all the tests and results executed using our simulation tool. Chapter seven, "Conclusions" presents the conclusions reached at this work and the theories derived from them and Chapter eight, "Future work" gives some insights on several different directions that were out of the scope of this thesis but may lead to some very interesting results based on the work done in this thesis.

2. Related work

2.1 Simulations of Transportation Networks

Transportation networks simulations are divided into two groups: macro simulations and micro simulations. Macro simulations [1-5] mimic traffic behavior by using mathematic calculations of flows, while micro simulations simulate the movement of individual cars. Micro simulations models can be divided into a few groups that differ from one another by each one's level of fidelity [6]. These groups are referred to as CF, CA, CTM and the queue model. CF (Car Following) is the most accurate model [7-9]. It uses realistic driver behavior, detailed vehicle characteristics and includes a complicated acceleration and deceleration algorithm. The problem with this model is that even with today's advanced computers, the simulation, using many cars, takes more time than it would in a real world transportation network. The next model, the CA (Cellular Automata) is much faster [10-11], yet it is still a high resolution model. Each road is divided into small cells that allow only one car at a time. In CTM (Car Transmission Model) [12], the cells are much larger and contain many cars. It is not necessary to know where the cars are located in the cell because the traffic flow is calculated using more general rules. Finally, the fastest and simplest model is the queue model [13-16]. In this model the roads are represented by queues and are not divided into cells. The velocity and distance between the cars are not relevant to this model.

Our model is a combination of a few different models. On the one hand, a simple queue model is not adequate because the act of gossiping is done between passing cars, so each vehicle's location on the road is important. Yet on the other hand, a full CF or even a CA model is too heavy if we intend to support gossiping between every two individuals. So we propose a combination where each road is divided into small sections and the cars are moved from section to section. However unlike a CF or CA model, our model's velocity (and therefore the road's capacity) does not depend on the distance between the cars. Instead, each road also has a queue with a queue size,

and when a car has finished traversing a road, it is put into the queue until it is allowed to enter the next road.

Several micro simulation tools have been developed for transportation networks. Some of the tools, for example DYNAMIT [17] and DYNASMART [18] use a relatively simple model similar to the queue model while others like TRANSIMS [19] use a more complex model similar to the CA model.

2.2 Routing in Transportation Networks

When addressing the problem of traffic jams and congestion there are two kinds of solutions. The first is to build additional roads or improve the road infrastructure. The second is to use advanced technology in order to improve the navigation of the cars in the network. Such a solution is part of a wider infrastructure called ITS (Intelligent Transportation Systems) [20]. ITS systems combine humans, roads and vehicles using state-of-the-art IT technology in order to provide high quality services for transportation networks.

Many routing algorithms have been suggested for both communication networks [21-24] and transportation networks [25-30]. Some use basic algorithms like Dijkstra, Bellman-Ford and A* to calculate the shortest path while others use advanced techniques like Genetic Algorithm and Reinforcement Learning.

Our work focuses on gossip networks. Therefore the investigation of the routing problem is done in order to learn the effects of the gossip network characteristics on the quality of the routing results. This is done by comparing the routing results in gossip networks to the results in other information layers like centralized information. For this purpose the specific algorithm used to calculate the shortest path is less important. What is much more important is that the same algorithm will be run for all the different networks in order to make a comparison between them. Therefore the Bellman-Ford algorithm was chosen for calculating the shortest path.

The routing concept used in our work is different from that which is used in other transportation networks. In many of the routing algorithms the goal is to optimize the network as a whole, to reduce traffic jams or to reduce the average travel time. Our goal is to optimize individual cars even on the account of optimizing the network. Furthermore, even in the research where the goal is to optimize individual cars the routing algorithm is executed by external routers that send the results to the cars on

demand. In our case each car is totally independent and calculates the routing algorithm by itself.

2.3 Gossiping

In the literature, gossiping [31-33] is known as a method of distributing information between individuals in a network. In contrary to broadcasting, where one individual broadcasts to all the others, gossiping enables every individual to pass some information on to his local environment. Gossiping has been studied in general graphs and in communication networks but rarely in transportation networks. Moreover, the method of gossiping used in this work is significantly different from the gossiping mentioned in the literature. While gossiping in communication networks is invoked by the network nodes (or routers), in our case it is done by the cars traveling in the network. The cars are mobile and therefore their movement influences the gossip behavior. The concept of using ad hoc communication between cars in transportation networks in order to pass information regarding the network itself was already introduced in [34] and the concept of "gossip networks" was first introduced in [35]. There are even some projects like CarNet [36] and FleetNet [37] that implement ad hoc communication between cars using radio transmission. But an investigation of the characteristics of gossip networks and their influence on the quality of routing in the network is first introduced in this work.

3. Gossip Networks

“Gossip Networks” introduce a new kind of network model that is composed of two layers - a physical layer and an information layer. The physical layer describes the structure of the physical network, edges, nodes and the relations between them. The physical layer in gossip networks is the same as in dynamic networks. In dynamic networks the edges contain weights that vary in time. The information layer describes how information about the network is distributed in the network. In gossip networks such information is distributed in the network using a unique gossiping model. In the gossiping model the agents gather information stored on the edges while traversing them. This information is further distributed in the network when two agents passing one another at close range exchange information between them. The information is passed in two ways: when cars on opposite roads pass each other or when cars standing at a junction exchange information.

3.1 Gathering Information

In order to perform an act of gossiping, a special device is required to be installed in each car. This device will enable the agent to gather information and communicate with other (neighbor) agents in the network. (The internal architecture of this device is not in the scope of this paper.) Furthermore, each agent has an internal database where he stores the most recent information about all the edges in the network. For every edge, the agent stores the weight and the time it was gathered. Every time an agent traverses an edge he updates his database with the new information about the edge.

3.2 Exchanging Information

Each time two cars exchange information by gossiping, they pass each other a portion of the information stored in their databases. The new information is updated in the receiving car's database only if its time is more recent than the time which is currently in its own database. The amount of information traveling in the network mainly

depends on two factors: the percentage of cars participating in the gossip process and the amount of information exchanged in each act of gossiping.

3.2.1 Gossip Car Percent

In a real transportation network it is not likely that all the cars traveling in the network will be able to perform gossiping. Therefore we need to define the "Gossip Car Percent" which means the percentage of cars in the network that are participating in the gossip process.

3.2.2 Information exchange

The cars exchanging information might be passing one another at great speed. So the wireless communication between them will probably not be fast enough to exchange all the information in their databases. Therefore we need to define the "Information Exchange Rate" in order to limit the amount of information exchanged in the act of gossiping.

It is reasonable to assume that the higher the gossip car percent and the information exchange rate, the more information will be traveling in the network. But how much more information will be traveling in the network and how will it affect the probability of receiving this information? These questions will also be explored in this paper.

3.3 Information Expansion

Thanks to gossiping, when an agent traverses an edge he may gather not only information about the edge traversed but also about any other edge in the network. This causes a phenomenon of information spreading or information expansion over the network. In order to quantify the information expansion we will define two measurements associated with every pair of edges (N,M) in the network. The two measurements are the "Edge Information Average Time Between Updates" (EIATBU) and the "Edge Information Delay" (EID), both of which are defined below.

Note that although the information is transferred by the agents the measurements refer to the edges.

3.3.1 EIATBU measurement

The EIATBU (Edge Information Average Time Between Updates) measures the average time between updates received by M about N. The EIATBU value is calculated by the formula

$EIATBU = \text{Simulation_length} / \text{number_of_updates}$. For example, if M received information about N five times during the simulation and the simulation length was 20 steps, then the $EIATBU(N,M)$ value will be $20/5 = 4$.

3.3.2 EID measurement

The EID (Edge Information Delay) measures the average delay of the information from N to M. The delay is defined as the difference between the time the information about N was gathered, to the time it was received by M.

For example; if three different cars traversed the edge N at times $t_1=8$, $t_2=11$ and $t_3=15$ and an additional three cars, c_1 , c_2 and c_3 (they can be the same cars or different cars) containing this information (c_1 containing t_1 and so on) traversed edge M at times $u_1=20$, $u_2=17$, $u_3=22$ (c_1 at time U_1 and so on) then the EID will be $((u_1-t_1)+(u_2-t_2)+(u_3-t_3))/3 = ((20-8)+(17-11)+(22-15))/3 = 9$.

3.4 Exploring characteristics of Gossip networks

Potentially there are many factors that may affect the EIATBU and EID values.

Some of them are:

1. The distance between the edges
2. The traffic load on the edges which have been measured
3. The Gossip car percent
4. The Information exchange rate

The results later in this paper will show which of these properties really affect the EIATBU and EID and in what way.

4. Routing Guidance in Dynamic Transportation Networks

As already mentioned, we are using a transportation network as an implementation of a gossip network. The physical network is composed of roads connected by junctions. Each road contains a weight that is equivalent to the time it takes to traverse said road. The road's weight varies in time according to the traffic load. Each car is associated with a pair of source and destination points. A "journey" is defined as a path between a source point and a destination point. A "path weight" (or a "journey weight") is defined as the sum of the weights of the edges composing the path. Every car wants to optimize its journey by finding the path with the minimal weight between the source and destination points. This is not a simple task since the road weights vary in time and the cars only have partial information about the road's weights.

It is important to establish that we are treating the network as a noncooperative network [40]. In noncooperative networks agents make decisions that are aimed to optimize their individual performance objectives. In other words, we are not trying to optimize network performance, but rather to optimize each car's private path. So in no case will a car select a longer path in order to optimize network performance.

4.1 Routing Guidance Comparison

The routing problem is studied by comparing the same routing algorithm for different networks. The different networks are all dynamic networks but each have a different information layer. This way the gossip model can be compared to other information layers such as – the centralized network model or the decentralized network model. Although the same algorithm is run for several networks, its behavior is different for each network since the information available in each network is varied. Not all of the cars in a network have the same capabilities. In a centralized information model, only a portion of the cars receive central information, and in a gossip model, only a segment of the cars can gossip. In order to compare between the different information layers we define "Special Cars" as the cars in a network that have that network's

unique capabilities. For example, in gossip networks the Special cars have gossiping capabilities and in central networks the Special cars have central information. The cars that do not have the special capabilities will be called "Regular Cars". These cars are able to gather information about roads only when traversing them.

4.2 Types of Information Layers

The information layer describes how information about the network is distributed in the network. There are two approaches which are used to study how information can be distributed in the network. One approach is to use central information and the other is to use ad hoc communication. In the central information approach, the data about the network is centralized in one place and from there it is distributed to all the cars in the network. In our implementation of central information, we assume that updated information (about all the roads in the network) is always stored in the central location. The gossiping model studied in this work is an example of the ad hoc communication approach.

We will be comparing four types of information layers:

1. Decentralized Information Model – No Special cars, all cars gather information about roads only when traversing them
2. Gossip Model – Some of the cars are Special cars with gossip capabilities and the rest of the cars are Regular cars
3. Centralized Information Model – Some of the cars are Special cars that have central information. The central information is available to them only when entering the network and includes the current load/weight of all the roads in the network. The rest of the cars are Regular cars.
4. Online Centralized Information Model - Some of the cars are Special cars that have central information. They are updated about the network state every time they reach a junction. This way they have the most updated information about the whole of the network every time they need to make a navigation decision. The rest of the cars are Regular cars.

4.3 Routing Algorithm

The routing algorithm is based on the Bellman-Ford shortest path algorithm for calculating the shortest path between the source and destination points. The Bellman-Ford algorithm is invoked every time a car receives new information regarding the roads' weights. However when all the cars invoke the exact same algorithm the network behavior becomes peculiar and irregular. The problem is that many cars select the same sub path even though there are many alternate sub paths with nearly the same or even exactly the same weight. In order to overcome this problem a few improvements have been added to the algorithm:

1. The algorithm is probabilistic in order to produce different results for different cars. So if there are a few paths with the same weight the cars will divide between them.
2. Each time a car is supposed to recalculate the shortest path, it throws a dice and only at some probability does it recalculate the shortest path.

Although the Bellman-Ford algorithm provides the shortest path available given the information the car has, it will seldom be the actual best path since the agent usually has only partial information about the weights of the roads.

The same routing algorithm is used for all of the types of information networks. Even so, the behavior of the algorithm will be different from network to network. The reason for this is that the number of times the Bellman-Ford algorithm is invoked and the qualities of its results depend on the amount of information available. The amount of information available is of course, different from network to network.

As mentioned above, every time (with some probability) a car receives new information it will calculate the shortest path between its current point and its destination using the Bellman-Ford algorithm.

New information is received in one of three ways.

1. When traversing a road – the information about the road's weight is gathered
2. Jammed road situation - if entering another road is not possible because it is jammed, this information is available to the car at the junction even though it hasn't traversed the road yet
3. Information received from gossiping or from central information

A Regular car can only receive new information in the first two ways. Since the information about a road already traversed is not helpful in recalculating the shortest path, the only situation in which a regular car will calculate the shortest path is when entering the network or in a jammed road situation (the second way).

On the other hand, Special cars may recalculate the shortest path many times during their journey. In gossip networks, this recalculation will occur every time new information is received by gossiping, and in central networks it will occur every time central information is received. So although the same algorithm is always used, its behavior will alternate for different cars in different networks.

4.4 Routing Performance

How should the routing performance be calculated?

Every car has a "journey time" which is the time it took to complete the journey.

One method of calculation might be to figure the average journey time for all the cars in the network. Another method could be to calculate the average network load. But both these measurements apply to all the cars in the network while only some of the cars in a network are Special cars that have the qualities of the specific network.

Therefore it may be more interesting to calculate the average journey time only for the Special cars.

Which network should have the best performance?

In the online centralized information model, all the information is always available- therefore it should have the best performance. In the decentralized information model there is no information known except for the data gathered by the car itself- therefore its performance should be worse. The gossip information model and the centralized information model should rate somewhere in the middle. While the centralized information model has the advantage of receiving information about all the roads, this occurs only once during the journey. In the gossip information model, only a portion of the information is received but it can be received several times during the journey.

4.5 Special cars percentage

The Special cars' percentage has a major effect on the routing guidance problem. In gossip networks the cars themselves are responsible for the distribution of information regarding the network state, so an additional number of Special cars should lead to a higher expansion of information and therefore to a better routing performance. In centralized networks all the cars are constantly updated with all the information about the network state, so seemingly the Special cars' percentage should not be relevant.

But as will be proven later in this paper (chapter 6) the Special cars' percentage is very significant. In centralized networks and even in gossip networks the effect is not as straight forward as expected.

5. Simulation

Due to the complexity and dynamics of the problem, the investigation has been performed via simulation. There are quite a few transportation simulations out there but none of them met our needs. This is mainly because our gossip model required the ability to freely exchange information between individual cars. Therefore a special micro simulation tool was built in order to support our special needs.

5.1 Simulation structure

The simulation was written in C++ and can run on a single PC. The simulation code is composed from two modules, the simulation engine and a user interface. The two modules are completely independent and communicate with one another through a well defined interface. Because of this separation, any part can be exchanged without having an effect on the other. For example, the user interface can be upgraded to a more advanced graphical application without touching the simulation engine. The simulation tool keeps all its information in configuration files so the network can either be created through the GUI or can be created elsewhere and then loaded into the application where it can be modified. The simulation can also save its current situation at any point so that the simulation can be run many times from the same point.

5.2 Simulation model

Transportation simulations are divided into two groups: Macro simulations and Micro simulations. Macro simulations mimic the traffic behavior by mathematic calculations of flows. Micro simulations on the other hand, try to resemble traffic behavior by mimicking the movement of individual cars. To support our gossiping needs, we required the use of micro simulation.

In micro simulations there are two implementation options.

The first option is to divide every road into small sections and move the cars from section to section. In addition there is a need to calculate the distance between the cars and its effect on the cars' velocity and the roads' capacity. The second option and the more simple one is to use the queue model. In this model, each road is represented by a queue with a queue size, and the cars are moved from queue to queue.

Our model is a combination of the two options. A simple queue model is not sufficient because the act of gossiping is performed between passing cars, so every car's location on the road is important. On the other hand we didn't want to monitor the distance between every two cars and calculate its effect on the cars' velocity and the roads' capacity, because this would require a complicated and inaccurate process. So we propose a combination of the two options, where each road is divided into small sections and the cars are moved from section to section. However, their velocity and therefore the roads' capacity do not depend on the distance between the cars. Instead, each road also has a queue with a queue size and when a car has finished traversing a road it is put in the queue until it is allowed to enter the next road. The cars are added to a road queue until a specific queue size is reached. When this occurs, the road is blocked from incoming cars until the queue size is decreased. Limiting the size of the queue has a very important impact called the "backwards traveling kinematics wave" (also known as the "jam wave"). This means that if there is a traffic jam on a specific road, the roads leading to it will also be influenced by it. Of course the bigger the traffic jam, the bigger its influence on other roads.

5.3 Physical Network model

The network is composed of roads and junctions. Each road is divided into sections. The cars advance from section to section until they reach the end of the road. Then they are put in the road's queue.

A section is the basic unit of length. A section is defined as the distance a car proceeds in the minimal velocity in one step (see movement model for definition of step). The time it takes to traverse a road is therefore $(\text{length} / \text{velocity}) + \text{time spent in queue}$, where the time spent in the queue is not known and depends on the traffic load (see movement model below for more details). For example: a road length is 10 sections. Two cars, car1 and car2 are traversing the road, car1 with velocity 1 and

car2 with velocity of 2. The time it will take car1 to traverse the road is $10 / 1 +$ time spent in queue. Car2 has the velocity of 2 and will advance two sections per step so the time it will take him is $10 / 2 +$ time spent in queue.

Each road has the following properties :

length – the number of sections the road contains

velocity – the velocity one can drive on this road, the range being between 1-5

junction_pass_size – defines the number of cars that can exit a road on every iteration of the traffic light

queue size – the number of cars allowed to be in the queue. After this number is reached the road is blocked.

5.4 Movement Model

The simulation minimal time resolution is described by the unit "step". A step is defined as the time it takes to pass a road's section at the minimum velocity. Every time the simulation clock is advanced by a step all the cars are advanced according to their paths. The cars located in the middle of a road are moved to a section ahead (how far ahead depends on their velocity). The cars located at the end of the road are inserted into the road's queue. The cars in the queue are moved to the next road only if they fulfill the terms of the junction.

Junction activity – On every step each junction pulls cars from its incoming roads and places them on one of its outgoing roads that matches the cars' paths. It does this according to the junction rules:

1. The number of cars pulled from a road will not exceed the junction_pass_size property of the road.
2. A car will only be pulled if it is the first one in the queue.
3. The car will only be pulled if the outgoing road matching the car's path is not blocked.

5.5 Traffic model

There are a few factors that affect the traffic load of the network:

- a) The number of cars inserted into the network
- b) The source and destination of each of the cars

c) Incidents like accidents that slow down the traffic at certain points

In our simulation these properties of the traffic are defined by rules that are executed during the simulation. In every step of the simulation all the rules that are in the time zone of that step are executed.

Car rule

Every car inserted into the network is associated with a source and destination point (see chapter 4 above for more details). Each Car rule determines how many cars with the same source and destination pair, will be inserted into the network in every simulation step in the specified range.

A Car rule contains the following information:

number_of_cars – number of cars to insert per step

source_node – the node in which the car is inserted to start its journey

destination_node – the end node for the car's journey

start_time – the start time in the simulation when this rule is relevant

end_time - the end time in the simulation when this rule is relevant

The simulations that were run in our tests contain between 70 and 100 different car rules for each simulation (see Tests & Results for more details).

Traffic rule

A Traffic rule creates an incident which limits the number of cars that can enter a road at every step (junction_pass_size). A traffic rule contains the following information:

road_ID – the ID of the road where the incident will occur

start_time – the starting time of the incident

end_time - the end time of the incident

incident_intensity – the gauge which indicates to what degree the junction_pass_size should be limited

In every simulation there are between 2 and 4 different traffic rules (see Tests & Results for more details).

5.6 Simulation Configuration

The network chosen is an illustration of the central part of Jerusalem, the capital city of Israel. The network contains about 50 junctions and 150 roads which is approximately the number of main streets that exist in the center of Jerusalem. The simulation length is 700 steps. Each step is equivalent to about 30 seconds in a real

transportation network so the simulation length is about six hours, which is a bit less than a full working day. The number of cars passing through the network during the simulation is about 70,000 and the average number of cars in the network at a specific time is 3,500 cars. The average length of a car's journey is 20 minutes.

5.7 Simulation Modes

The simulation can be run in two modes, "Single Iteration" and "Multiple Iteration". The single iteration mode resembles a single day scenario where the cars drive from a source to a destination, for example from home to work or the other way around.

In this scenario the cars only go through the network once.

In the multiple iteration mode the scenario is a sequence of several days, each day being equivalent to a single iteration. In every day of this mode, the cars drive from the same source to the same destination, except that on the following day, they have the information that was gathered the previous days. According to this information the cars can calculate a new shortest path.

Because the information from the previous day may not be relevant to the next day, the cars keep a history which takes into account the information from the past. The proportion between the history and the last information (from the previous day) is determined by the learning factor. The learning factor determines how much weight the information from the last day carries, in relation to the previous days. For example, if the learning factor is 50% then the current weight is the average between the weight from the last day and the history. This way if the traffic load is continuous and therefore relevant to the future it will have a major effect but if it was only temporary it will not affect the car's future path.

5.8 Simulation Stages

The simulation process is composed of several steps:

1. Creating Car rules – Since the traffic model is based on rules (see "Traffic rules") these rules must be created before the simulation is run. At this stage only the "Car rules" are created. Because the simulation needs to be run many times, many permutations of car rules need to be created. Therefore a process

is run which randomly produces many permutations of car rules. Although the rules are created randomly they still have to obey some conditions:

- a) The rules must be different from one another
 - b) The number of paths starting from a specific junction must not exceed a predefined limit
 - c) The number of paths ending at a specific junction must not exceed a predefined limit
2. Warm Up run – At the beginning of a simulation run the network is empty and the cars have no information about the network except for the physical distance of the roads. This situation does not mimic a real transportation network and therefore is not interesting. To avoid this situation there is a warm up stage where the simulation is run until the network is balanced and reaches a steady point. In order to reach such a point it is not enough to run the simulation once (again, all the cars at the first run have no information about the network). After performing many tests we saw that the maximum iterations needed for the network to reach its steady point are seven. So each simulation is first run in the multiple iteration mode for seven iterations, and the current state is then saved to a file. Saving this stage means saving the internal database with its knowledge of the network for each car that passed through it.
 3. Creating Traffic rules – As in the first stage, a special process is run in order to create many permutations of traffic rules so that the simulations can be run on many different configurations. These permutations are also created randomly under several constraints based on the information gathered from the warm up stage. The main constraint is that the roads associated with the traffic rules must not be roads with very little traffic (the lowest 10%), otherwise the traffic rules will be insignificant. The other constraint is that the number of traffic rules for a specific run must be between 2 and 4.
 4. Running the simulation - After all the rules were created, the simulation is run from the point saved by the warm up stage. The simulation running in this stage is much closer to a real network. The cars already have information about the network that is based on the past and the simulation contains rules that mimic accidents. All the statistics and analyses made are based only on the information gathered at this stage.

6. Tests & Results

6.1 Gossip network characteristics

The first goal of the tests was to investigate the characteristics of gossip networks (see Chapter 3). In order to achieve reliable results that will be correct for gossip networks in general, the tests were run in many different configurations. The parameters influencing a configuration are as follows:

- a) Starting conditions - The number of cars inserted into the network and the source and destination of each of the cars (see Car rules)
- b) Traffic incidents - The properties of the incidents created in order to mimic traffic jams (see Traffic rules)
- c) Gossip percent - The percent of cars that have gossiping capabilities

According to these parameters the tests were divided into groups. A group is composed of 6 different scenarios, each with different starting conditions. For each such configuration, 5 different sets of traffic incidents were created. A group of tests therefore contains $6 \cdot 5 = 30$ different simulation runs.

Six different categories of gossip percents were chosen – 60%, 40%, 20%, 10%, 5%, 2%. For each such category, simulations (a group of tests) with the same 30 configurations were run. That brings us to a total of 180 simulation runs executed in order to gather information regarding the characteristics of gossip networks. The following information was calculated for each test group:

- a) The length of every path used in the network
- b) The time it took to traverse each path on an empty network
- c) The average time it took to traverse each path when no traffic incidents were entered into the network

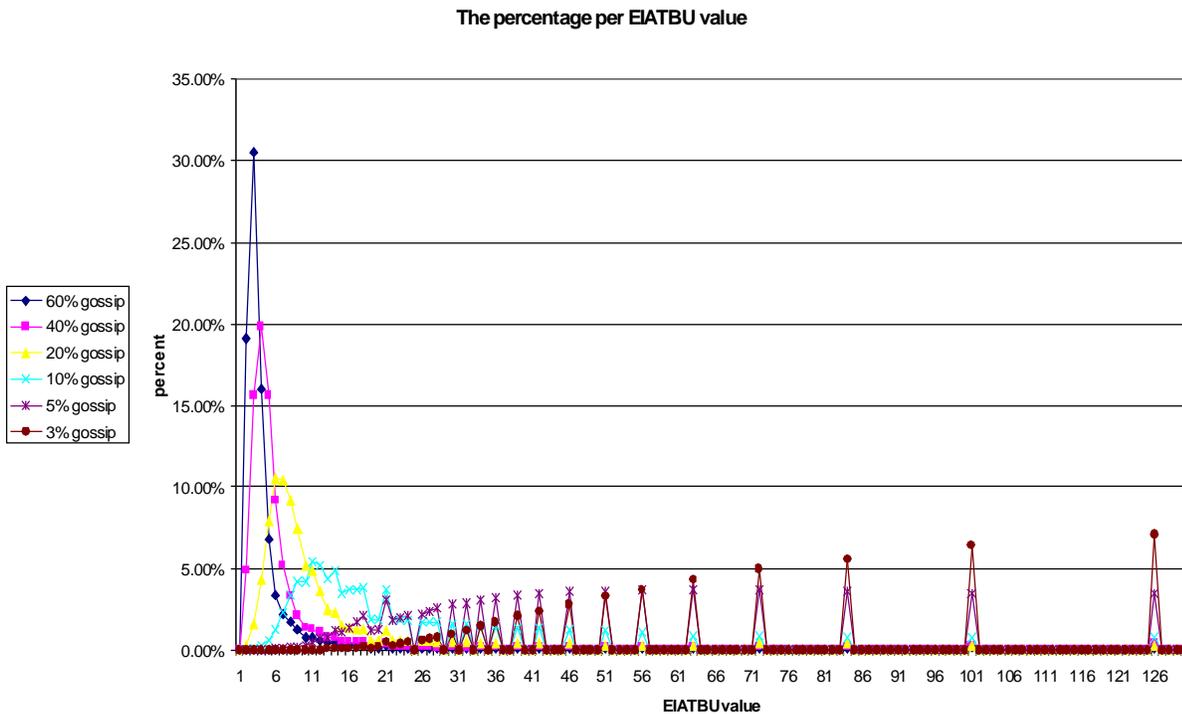
In addition, for each simulation run the following information was gathered:

- a) ETI - The number of cars entering each edge on every step
- b) EIATBU (see EIATBU measurement; chapter 3.3.1)
- c) EID (see EID measurement; chapter 3.3.2)

Test 1

This test measures the EIATBU value for each pair of edges/roads. The purpose is to understand at what frequency the roads are updated with information from other roads. The information is presented in a graph where the X axis is the EIATBU value and the Y axis is the percentage of pairs of edges that have that specific value of EIATBU. Each line in the graph represents a different gossip percent.

Results



From the results shown in Graph 1 we can learn the following things:

- a) The higher the gossip percent, the higher the update rate (smaller EIATBU value).
- b) From 20% gossip and up, most of the EIATBU values are concentrated between 1 and 10. Since the lengths of most of the roads are between 3 and 10, this indicates that for 20% gossip and up, usually an update accrues before a car manages to traverse the edge.

Note: The reason that high values of EIATBU are incremented in big jumps is because of the way the EIATBU value is calculated:

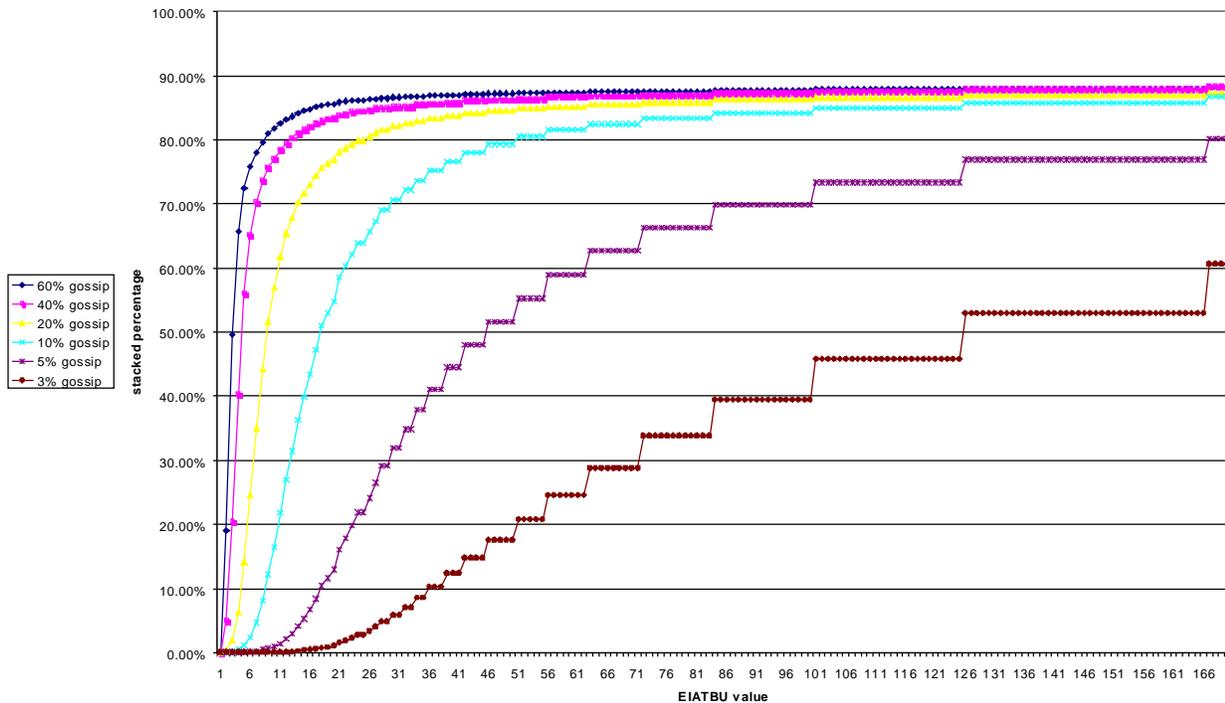
$EIATBU = \text{Simulation_length} / \text{number_of_updates}$. In our case $\text{Simulation_length} = 500$. So for the small values of number_of_updates , the EIATBU value is incremented in big steps, for example: $500/7=72$, $500/6=84$, $500/5=100$, $500/4=125$ (the graph is trimmed at 130).

Test 2

This test measures the accumulated percentage of the EIATBU value for each different value of gossip percent. The X axis is the EIATBU value and the Y axis is the accumulated percentage. Each line in the graph represents a different gossip percent.

Result

The stacked percentage of EIATBU values



From this graph we can see verification for the previous results (claim b in the previous graph) because for 20%-60% gossip we can see that the accumulated value at $EIATBU=10$ is 60%-80%, meaning most of the information is received before $EIATBU=10$.

Furthermore, we can see that for this range of gossip percent (20%-60%) about 85% of the total information available in the network is updated at least every 35 steps, which is the average time of a journey in the network. So on average, nearly all changes in the network state reach every edge in the network in a time less than the average journey of the cars.

Tests 3, 4

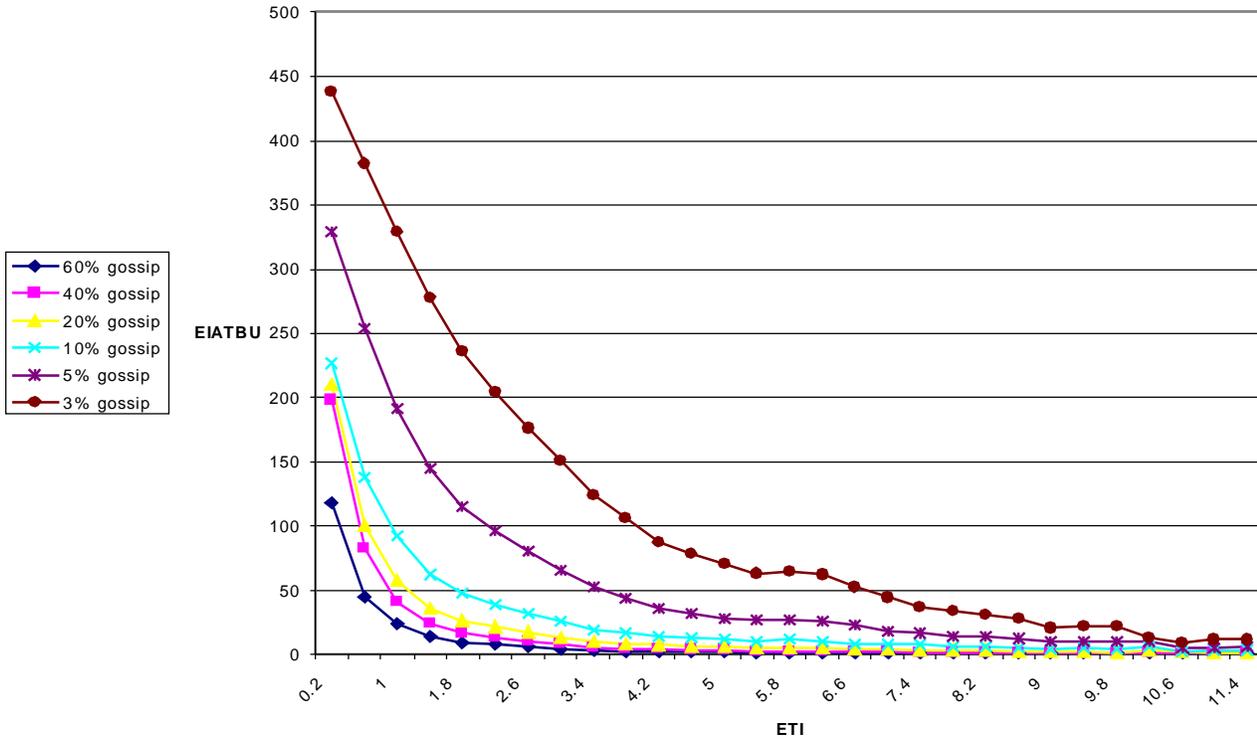
In the first two tests we saw the influence of the gossip percent on the $EIATBU$ value. The next two tests try to conclude which other parameters influence the $EIATBU$ and in what way.

Test 3

In the third test the connection between the $EIATBU$ value and the ETI value is studied.

The ETI (Edge Traffic Intensity) measurement is defined as the number of cars entering each edge on every step. Since the $EIATBU$ value is associated with a pair of edges, the ETI value is actually the average ETI value of the two edges associated with the $EIATBU$ value. The results are presented in a graph where the X axis is the ETI value and the Y axis is the $EIATBU$ value.

The EIATBU as a function of ETI



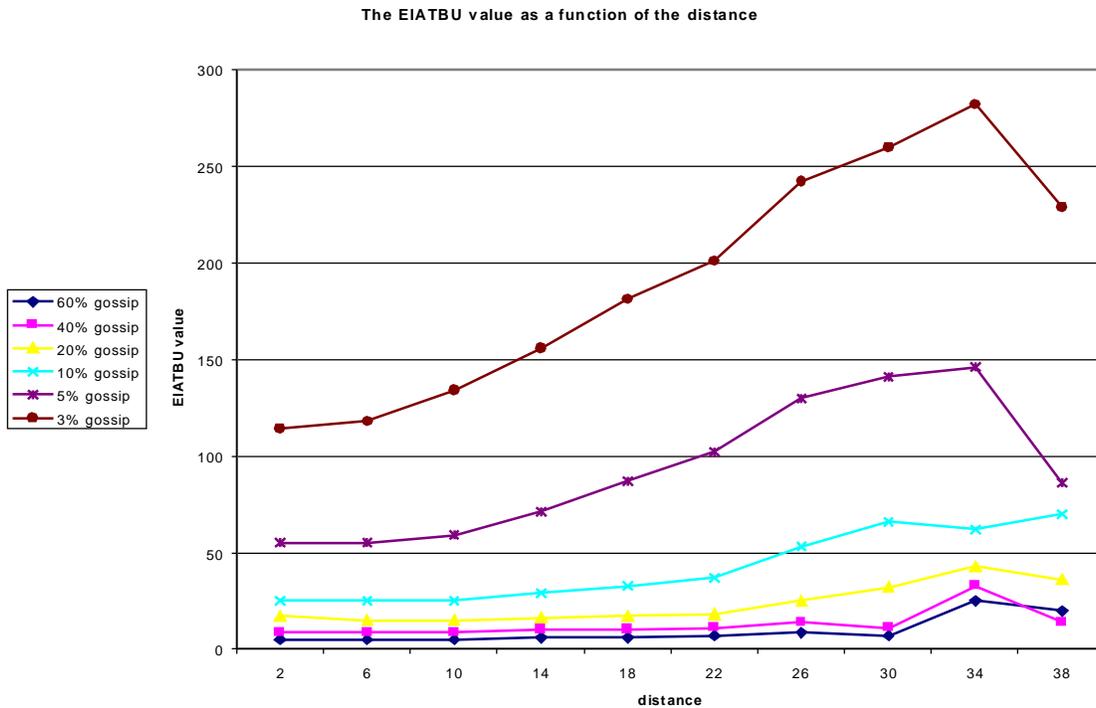
Results

The results show the high correlation between the ETI and EIATBU values for all gossip percent levels. The lower the ETI value is, the higher the EIATBU value is (the lower the update rate is). A higher ETI value promises that more cars were updated with the new edge information which in turn leads to a higher update rate of the information on the destination edge. For 20%-60% gossip percent we can see that an ETI value of 3.5 (3.5 cars entering an edge on every step) is enough to bring the EIATBU value to a range of 3- 10 which, as mentioned, is the average length of a road.

Test 4

This test studies the connection between the EIATBU value and the distance between the two edges associated with the EIATBU value. The results are presented in a graph where the X axis is the distance and the Y axis is the EIATBU value.

Results



The results show that in general, for the high levels of gossip percent there is no correlation between the distance and the EIATBU value, while for the low levels of gossip percent a bigger distance brings a lower update rate (a higher EIATBU value). This means that if the information flow between the two edges is constant, the distance does not affect the EIATBU value. This is as it should be because the EIATBU measures the update rate and not the information delay. But when the information flow is not constant, the distance also affects the update rate.

Tests 5, 6

These tests investigate the RGLP measurement. The **R**oad **G**ossip **L**earning **P**robability measures the probability to learn new information on every road. The

probability for learning new information while traversing a specific road is calculated as follows: every car, when traversing a road, counts the number of roads from which it received new information while traversing that road. When exiting a road, every car updates the road with that number. At the end of the simulation each road calculates the average for all the numbers it received and divides it by the number of roads in the network. The result is the learning probability for that road. For example, during the simulation, three cars (car1, car2 and car3) traversed road R1. While traversing R1, car1 received information regarding R3 and R4, car2 received data regarding R3, R4 and R5, and car3 received information regarding R4, R5, R6 and R7. When exiting R1 they will update R1 with the numbers 2, 3 and 4 respectively. Let's assume there are ten roads in the network. The learning probability on R1 will therefore be $(2+3+4/3) / 10 = 30\%$.

$t_old(road1)$ is defined as the time updated in the car's database for road1.

$t_new(road2,road1)$ is defined as the time associated with the information received on road2 about road1.

We defined two RGLP measurements based on two different definitions for "new information":

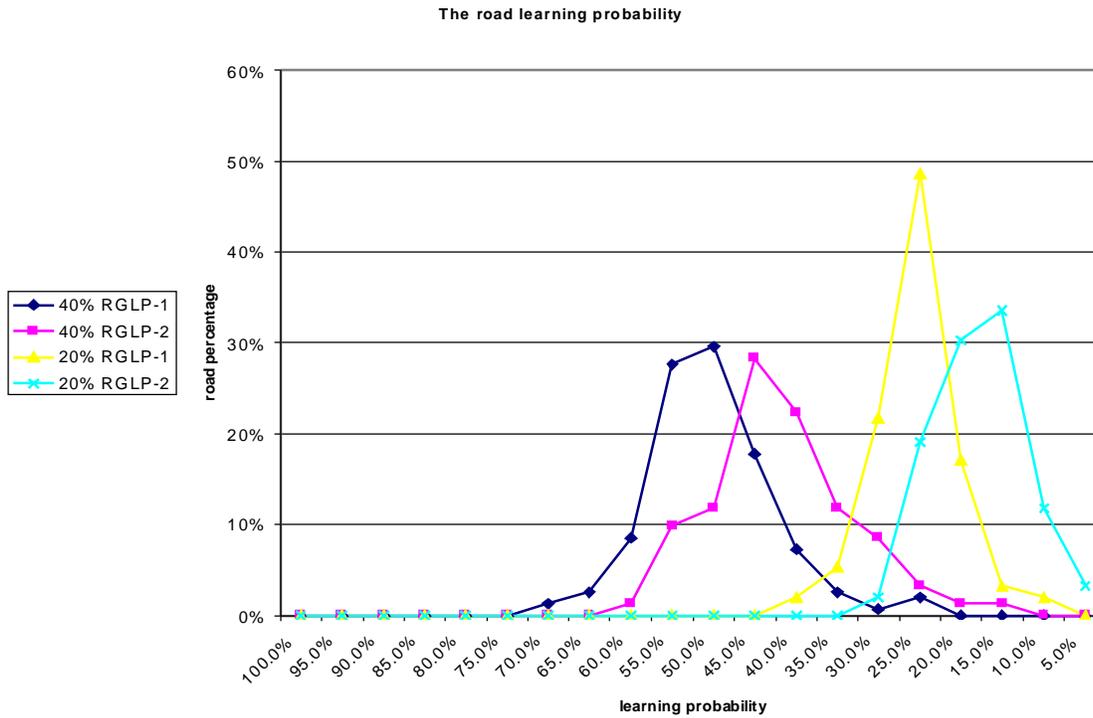
1) RGLP-2 - New information is received on road2 if $t_new(road2,road1) > t_old(road)$, meaning: any information received with time ("time" refers to the time the information is updated for) that is greater than the time already known to the car regarding that road, is considered new information.

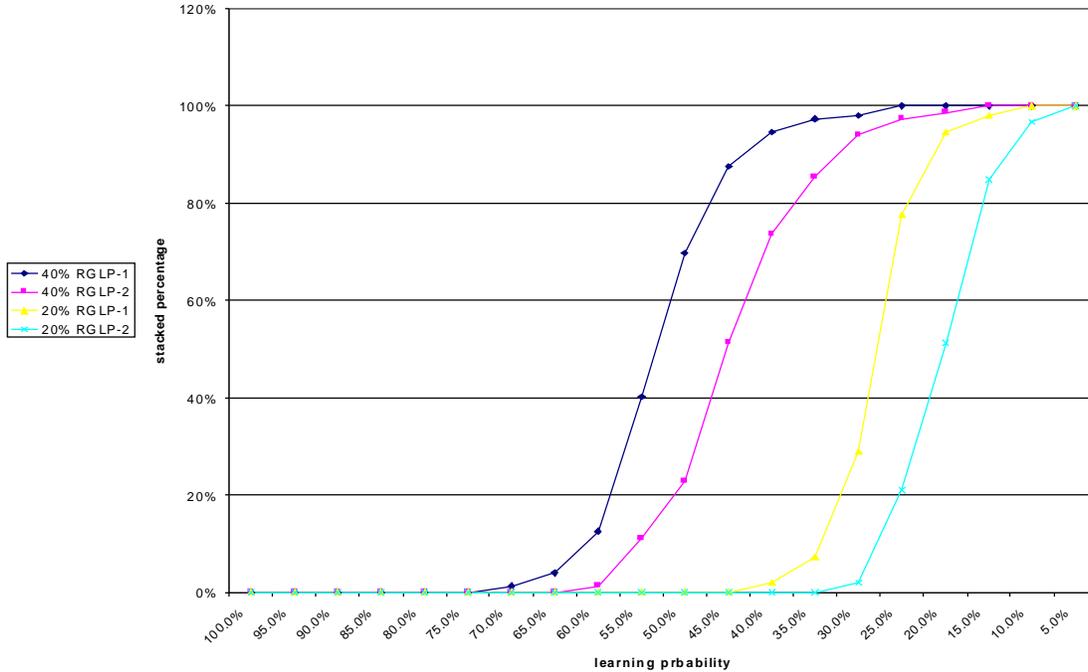
The problem with this definition is that the amount of new information received by a car on a specific road depends on its previous path. If the previous path contained information now available on the current road, then when received on the current road, this won't be considered new information-this data is redundant. To avoid this problem a second definition was given.

2) RGLP-1 - New information is received on road2 if $t_new(road2,road1) > t_old(road)$, meaning: any information received with time that is nearly as great or greater than the time already known to the car regarding that road, is considered new information.

The results are presented in two graphs. In the first graph the X axis presents the learning probability values and the Y axis presents the percentage of roads that had the corresponding learning probability. The graph contains four lines - two for each

measurement. For each measurement there are two lines - one represents the results for 20% of Special cars and the other represents the results for 40% of Special cars. The second graph presents the same information, except that the Y axis presents the stacked percentage starting from 100% and down.





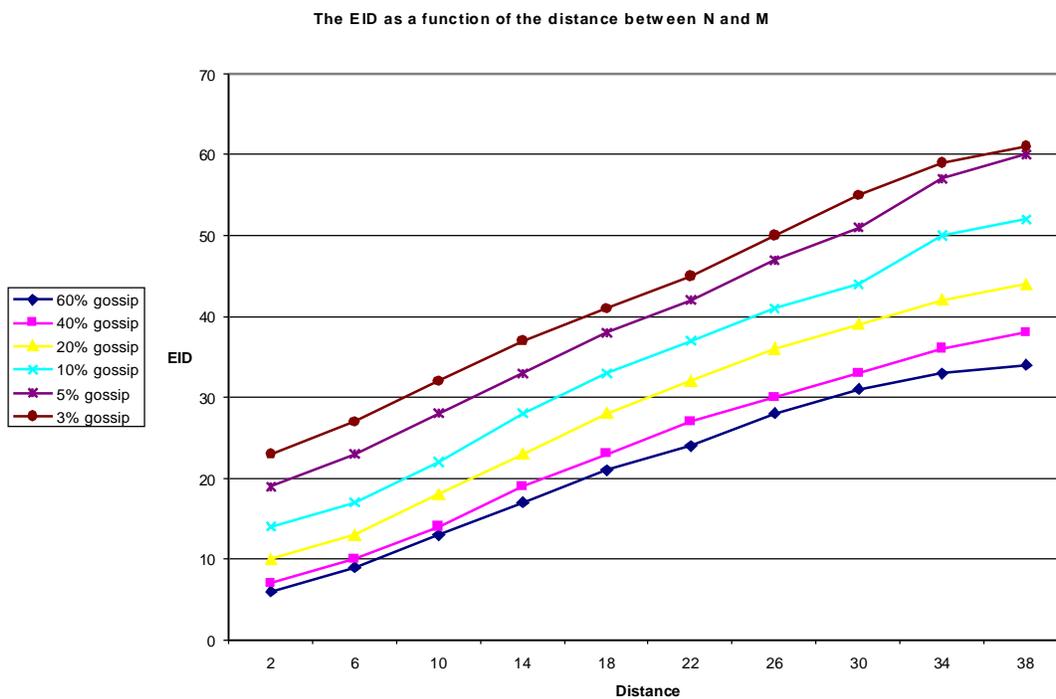
Results

The RGLP measurement is closely related to the EIATBU measurement (test 1, 2) because a high update rate of the information on a road should lead to a high learning probability. Indeed the results of the RGLP measurement match the results of the EIATBU measurements – in the second graph we see that for 40% of Special cars, about 70% of the roads have a learning probability of at least 50%. This can be explained by the fact that for the same level of Special cars, about 70% of the roads have an update (EIATBU) rate higher than an average road length (test 2). Furthermore, for 20% of Special cars about 70% of the roads have a learning probability of at least 25%. This can be explained by the fact that for the same level of Special cars, about 70% of the roads have an update rate higher than twice the average road length (test 2).

Test 7

This test studies the connection between the EID value and the distance between the two edges associated with the EID value. The results are presented in a graph where the X axis is the distance and the Y axis is the EIATBU value.

Results



The results show the high correlation between the EID value and the distance for all gossip percent levels. This behavior is quite predictable because the EID measures the information delay and the delay at least partially depends on the distance. Therefore the test is another indicator of the reliability of the network behavior.

6.2 Routing guidance in Gossip networks

The second goal of the tests was to investigate the routing problem in gossip networks (see Chapter 4). Since the traffic load changes in time the cars need to change their routing paths in order to optimize their path times. The quality of the information available to the cars should affect the routing performance. The investigation was done by comparing the routing performance between four different information layers (see Chapter 4). In order to compare between the different networks the same tests were run on each of the networks. As performed in the previous test, many trials with different configurations were run here, in order to receive general results. The parameters influencing a configuration in this case were:

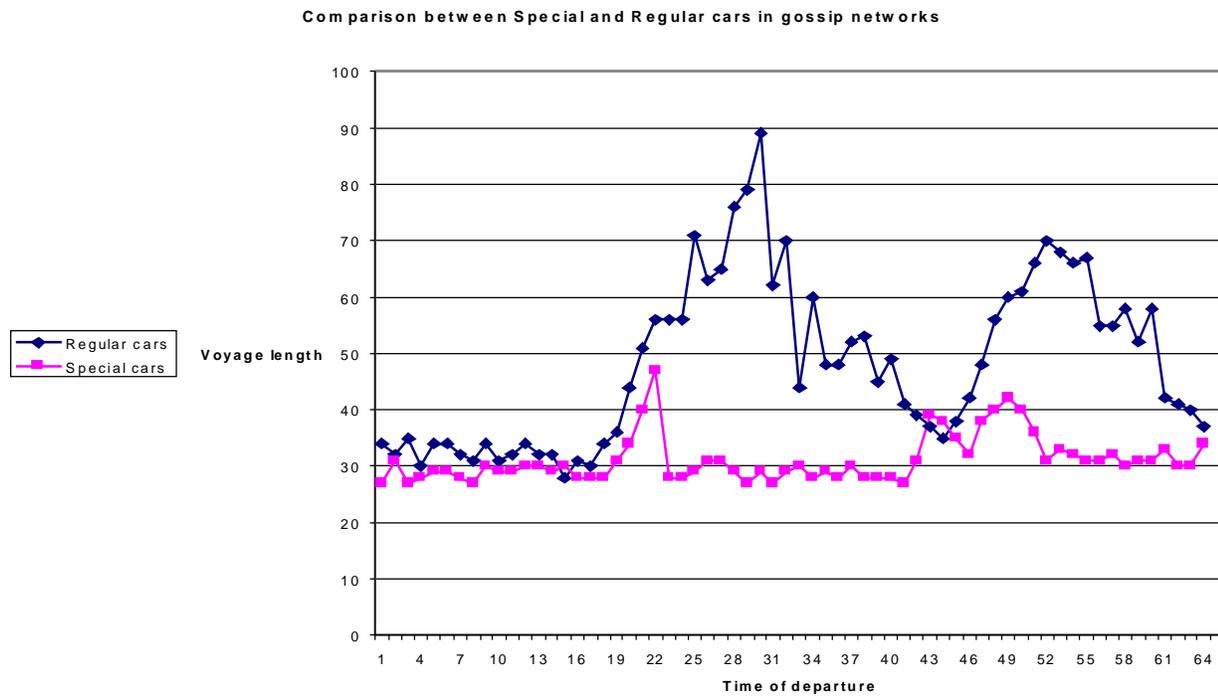
- a) Starting conditions - The number of cars inserted into the network and their source and destination (see Car rules)
- b) Traffic incidents - The properties of the incidents created in order to mimic traffic jams (see Traffic rules)
- c) Special car percent - The percentage of cars having the special network capabilities

The structure of the tests was identical to that of the previous ones, being 6 categories, each containing 30 tests, totaling 180 tests. The difference this time was that such a package of tests (180) was run for each type of network. So the total amount of tests executed for this stage was $4 \cdot 180 = 720$ tests.

An important comment regarding the tests in this section is that the results refer to all the cars in the network, while the traffic jams usually affect only a small portion of the entire cars in the network. The consequences are that an improvement in the routing performance (made by the cars avoiding a traffic jam) is less noticeable in the results because these cars are only a small percentage of the total cars.

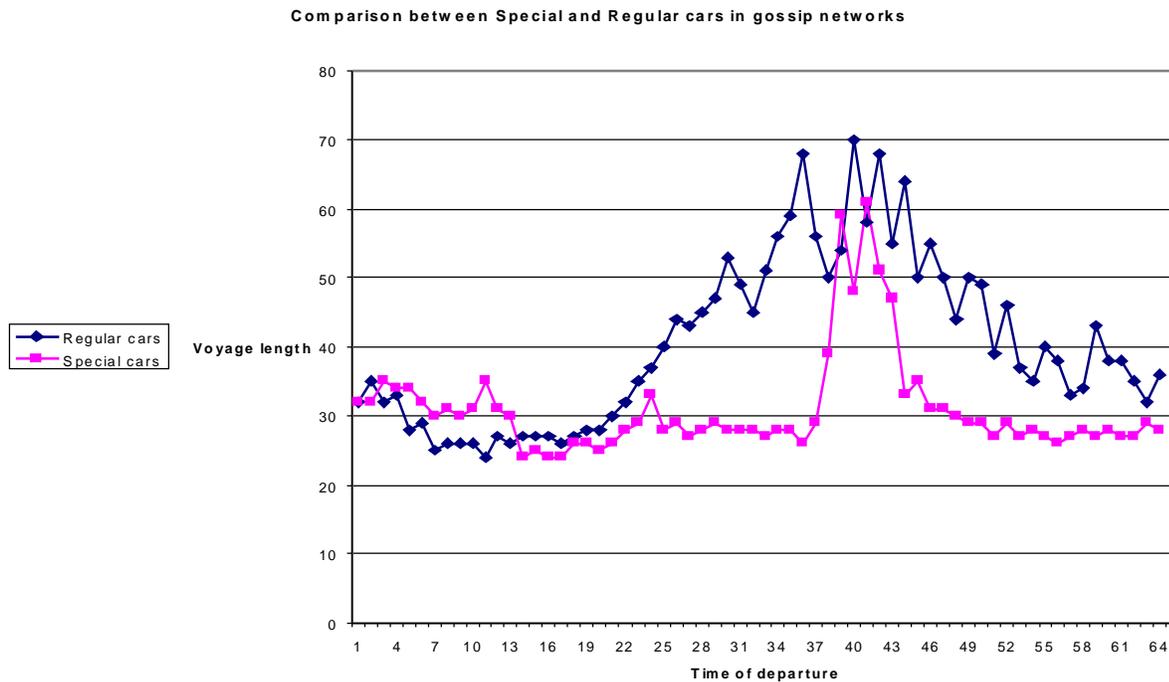
The first few graphs show several examples which compare the Special and Regular cars in gossip networks. Special cars are the portion of the cars traveling in the network that support gossiping, and Regular cars are the ones that don't (see chapter 4.1-4.2). In these examples the Special car percent is 20%, which is the number that leads to the optimum routing performance in gossip networks (see test 12). The graphs show the voyage time as a function of the time of departure. The X axis shows the time of departure, and the Y axis the voyage length/time.

Test 8



In the above graph you can see that after about 20 steps, several traffic jams occurred. This can be observed by the fact that the voyage length value increases gradually from that point. The fact that the Special cars line starts decreasing after a short time shows that the Special cars have "learned" about the traffic jam and are avoiding it. Notice that the Special cars line decreased to the original level, meaning they have completely learned the traffic jams. At about 30 steps the Regular cars line also starts gradually decreasing, meaning that the traffic jam is gradually decreasing. The reason for this is that Special cars are now avoiding the area of the jam, causing it to gradually clear. At about 45 steps the same process starts to repeat itself except that this time the Special cars are not able to completely recover from the jam.

Test 9



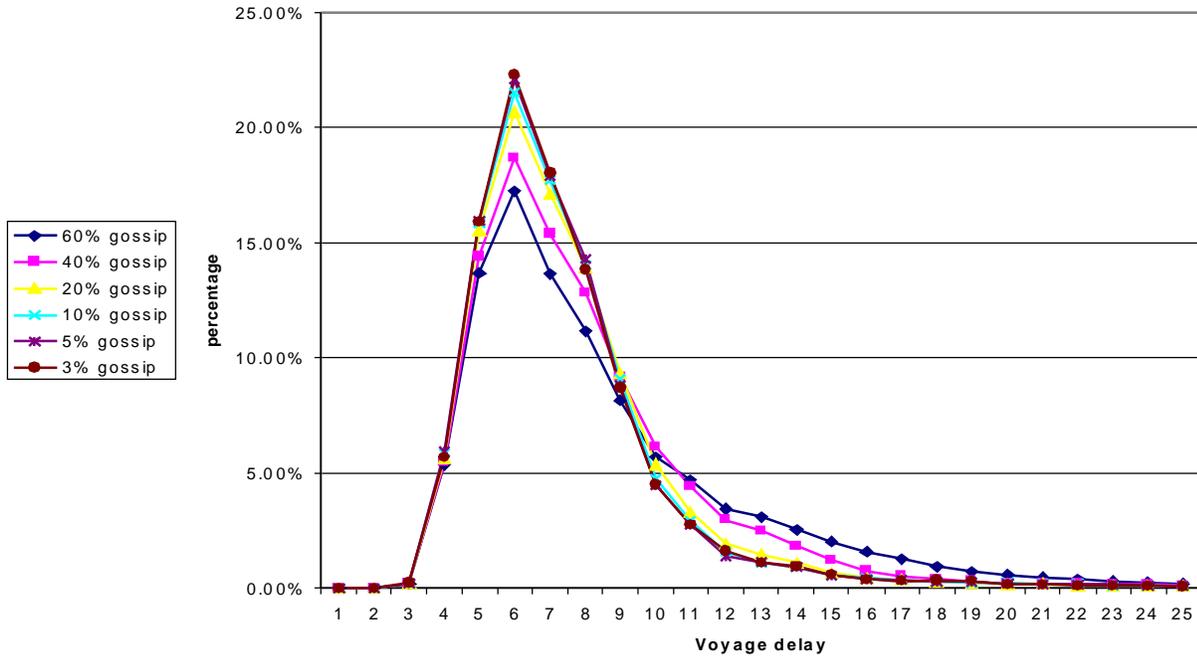
In the second graph you can see that a traffic jam is starting to evolve after 20 steps. The Regular car line increases gradually until step 40 and then gradually decreases. A strange phenomenon, seemingly, is that the Special car line not only decreases much faster (which is explained by the fact that the Special cars have "learned" the traffic jam- see explanation on previous graph), but also starts increasing at a much later stage. The explanation is that at the first stage the Special cars quickly learned the jam and only at a later stage when the traffic jams increased significantly, the Special cars line starts increasing significantly. A justification for this explanation can be found from the little peak at step 25 indicating that there was a slight increase in the Special car line at the beginning of the jam, but it was quickly overcome.

Tests 10, 11

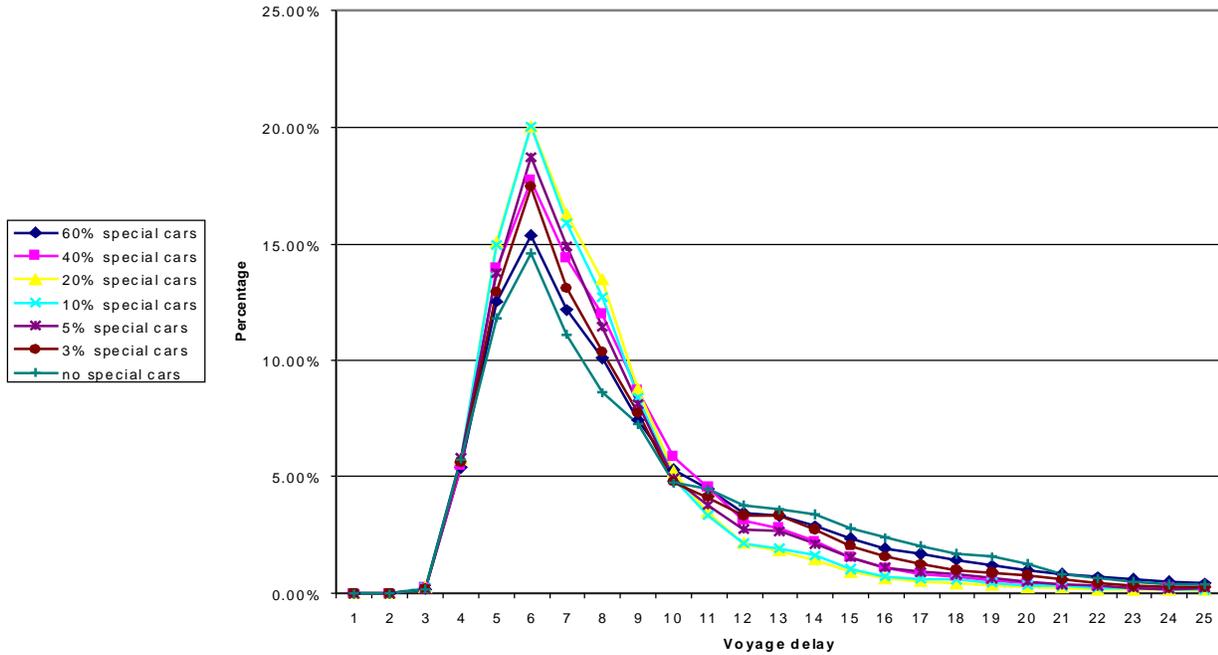
These tests compare six different levels of Special cars (see chapter 4.1). The comparison is for the voyage delay in gossip or centralized networks. The voyage delay for each car is the difference between the optimal path time and the actual path

time. The optimal path time is the time it takes to traverse the shortest path in an empty network while the actual path time is the time it took the car to traverse the path in the network containing all the traffic. The voyage delay value is presented according to the percentage of cars with that value.

The percentage per Voyage delay in Centralized networks



The percentage per Voyage delay in gossip network



Results

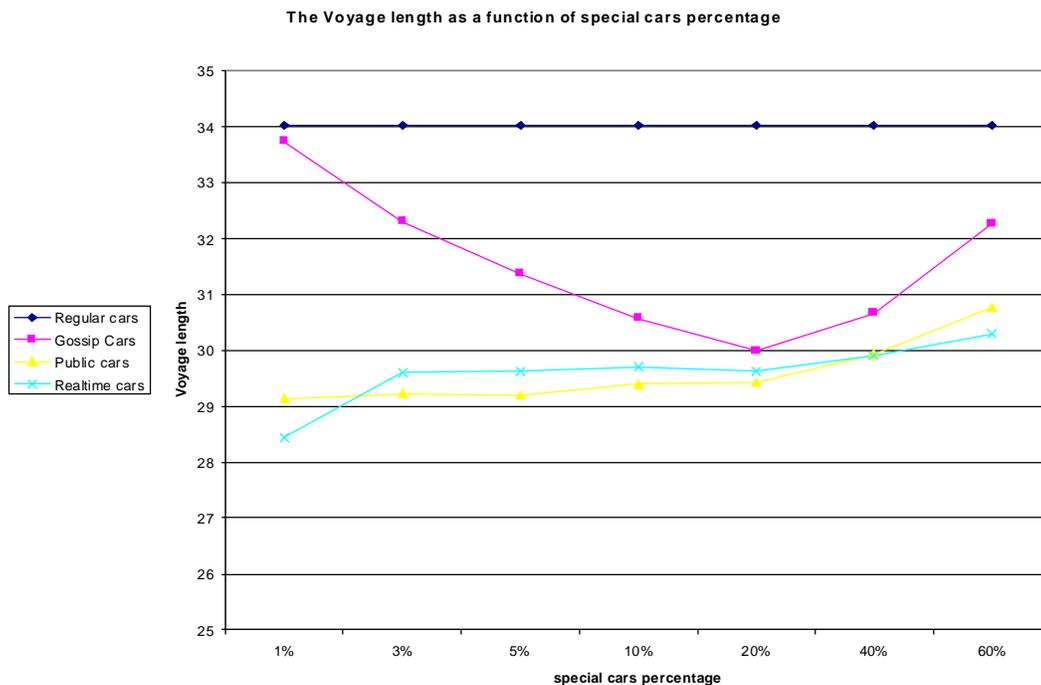
The first graph compares the voyage delay in centralized networks. In centralized networks each car is constantly updated with all the information regarding the network state (see chapters 4.1- 4.2). Seemingly this means that the percentage of Special cars isn't supposed to affect the voyage delay. Actually what we see from the graph is that it does affect the voyage delay- the higher the Special cars percent, the higher the voyage delay will be. The explanation is that more Special cars only disturb the other Special cars because when there are only a few Special cars they can select the best roads (empty roads) without anyone disturbing them. However, when there are many such Special cars that all want to select these roads, the selected roads themselves will become crowded and therefore less attractive.

The second graph compares the voyage delay in gossip networks. In gossip networks the Special cars depend on one another to gather information, due to the information being distributed by passing cars. So the natural behavior expected is that the higher the Special cars percent, the lower the delay will be. Here too, the results are different from what is expected. What we see from the graph is that the best results are achieved for 20% and 10% of Special cars and the worse results are achieved for 60% and 3% (except of course, for zero percent). The explanation for this order is that in

this case there is a combination of two contrary phenomena. The first phenomenon is the one explained above, regarding the fact that the cars need one another in order to distribute the information. From this point of view, a higher Special car percent is positive. The second phenomenon was brought above to explain the central network graph. The claim was that when there are many Special cars they disturb one another because they all select the empty roads and cause them to be crowded. The combination of these two contrary phenomena causes the optimum results to be achieved somewhere in the middle- at 20% of Special cars. Too many Special cars cause the second phenomenon to be dominant and too few Special cars cause the information distribution to be too slow. This behavior can be seen radically by the fact that in 60% of Special cars, the voyage delay is nearly as bad as in that of no Special cars at all.

Test 12

This test shows the big picture- the voyage length in each simulation is summarized to a single point and therefore a comparison between all four different information networks/layers (see chapter 4.2) is possible in one graph. The graph shows the average voyage length for the different levels of Special cars.



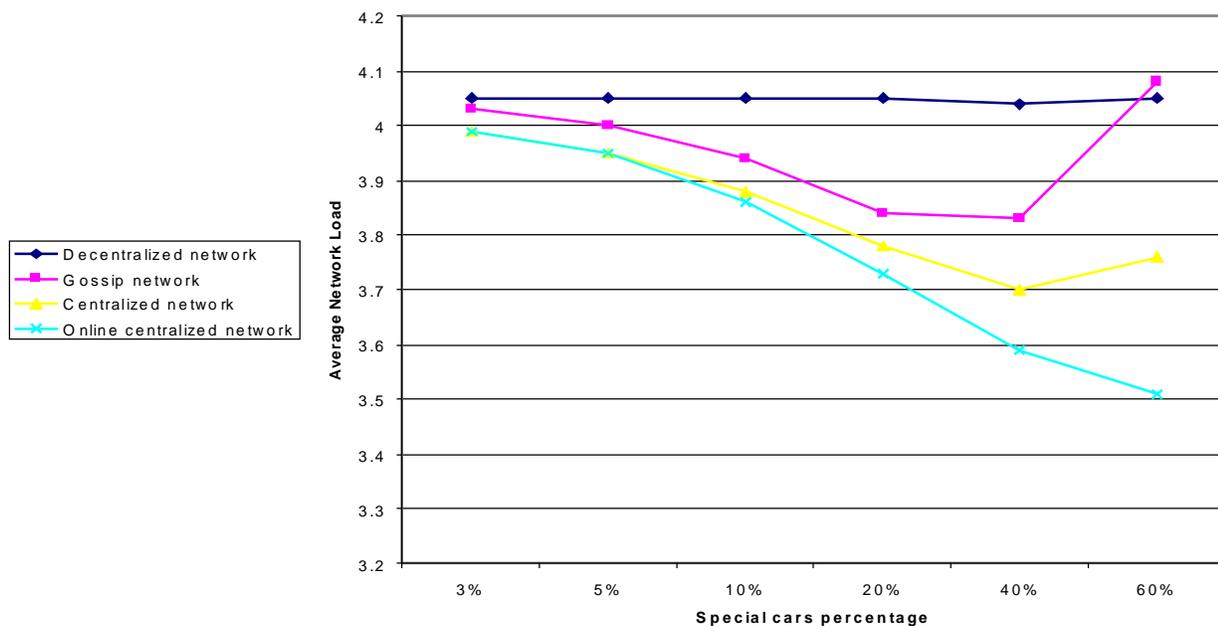
Results

The graph shows that the results for all the cars in the network are similar to the results only for the Special cars: for the gossip cars the optimum is at 20% and as you move from that point to any of the sides, the results get worse. For the centralized cars, as the Special cars percentage decreases the voyage length decreases. Another interesting point is that at the optimum, the gossip network is nearly as good as the centralized network (the gossip cars line is close to the public and real time cars lines at that value).

Test 13

The previous tests analyzed the routing performance by observing the voyage length parameter. This test gives a different perspective on the problem by measuring the average network load in each case.

The network average load per special cars percentage



Results

According to the results in the previous test we would expect that the network load value, like the voyage length in the previous graph, will decrease as the Special car percent decreases. The reason why we see a different behavior (nearly opposite) is

that the network load measurement takes into account all the cars in the network- not only the Special cars, while the voyage length measurement takes into account only the Special cars. Therefore while the Special cars are the majority, decreasing the percentage leads to an improvement in the network load (see previous test). But as the Regular cars' number increases they become the dominant factor and the network load increases because their performance is much worse.

6.3 Partial information

In the previous tests a basic assumption was made about gossip networks – that when two cars exchange information, they can exchange all the information in their databases. This is not necessarily a realistic assumption because the cars exchanging information might be passing one another at such great speed that the wireless communication between them probably won't be fast enough to exchange all the information in their databases (see chapter 3.2). Therefore in the following tests the amount of information passed between the cars was limited to 15 roads (which is 10% of the roads in the network). According to this limitation, the results regarding the routing performance in gossip networks were reevaluated.

When limiting the amount of information passed, a new question arises: What part of the information should be passed? Surely there is some information more important than other, but the level of importance may be different to different cars. In our tests we have defined three kinds of algorithms to decide which information to pass between the cars:

Fixed random - when entering the network each car will randomly select the subset of roads to exchange

Dynamic random - each time a car needs to exchange information it will randomly select a new subset of roads

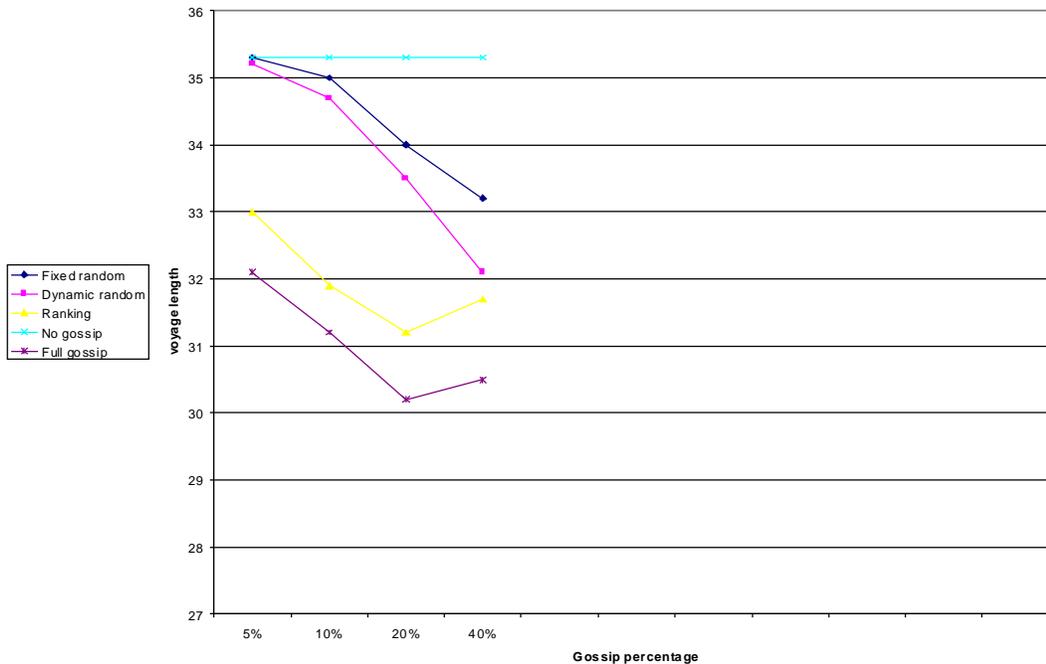
Ranking – each car will rank all the roads in its database and will exchange the highest ranked roads. The ranking algorithm is based on the assumption that the roads with the biggest change in their traffic loads are the most important ones because they would have the biggest effect on the routing algorithm. So every time a car receives new information it calculates the delta between the road's current weight and the weight known when entering the network. Then the road's new ranking is calculated as follows: $\text{new_rank} = \max(\text{old_rank}, \text{delta})$. The reason why the ranking can never

be reduced can be explained by the following example: The road's weight when entering the network was 10. At some stage of the simulation its traffic load is increased to 20. The car receiving the information updates its ranking to 10. Later on, the traffic load is reduced back to 10. If the car simply calculates the delta, the new rank will be zero. This can be problematic because there may be many cars which were already updated with the value of 20 and for them the return to 10 is worth the ranking of 10 and not zero. In the formula presented above, this problem is avoided. At the low levels of Special cars there shouldn't be a big difference between the fixed random and the dynamic random. But at the high levels when there are a large number of exchanges of information, the fact that a car exchanges a different subset of roads every time is much more effective and therefore the dynamic random algorithm should perform better. The ranking algorithm is expected to perform very well because usually there are only a small number of roads whose weight changes significantly. So if the algorithm is able to identify them, it should perform well.

Test 14

This test compares the voyage length between the three algorithms (presented above) for exchanging partial information, full information and no information (Regular cars).

The comparison is done at four levels of Special cars.



Results

We can see a big difference between the ranking algorithm and the random algorithms. The ranking algorithm is much better at the low levels of the Special cars percentage. The ranking algorithm, as expected (see introduction to the test), behaves well, despite the fact that it exchanges only 10% of the information. It behaves very similarly to the full information exchange - the optimum value is at 20% and then it rises on both sides. The random algorithms perform better as the Special cars percentage goes up. At 40% there is enough exchange of information in the network for the dynamic random to perform in nearly the same way as the ranking algorithm. The difference between the dynamic random and the fixed random is that the dynamic random is much more sensitive to the increase in the Special car percent as explained in the introduction to the test.

7. Conclusions

This thesis shows that the gossip network model is a surprisingly efficient information model despite the fact that the distribution of information is done only by exchanging information between cars passing one another. The cars participating in the gossiping process are able to detect traffic loads in a short time and bypass them by recalculating their pass to include new roads instead of the loaded roads (see test 8,9).

We show that in gossip networks there is a trade off when increasing the number of cars capable of performing the act of gossiping. On one hand, increasing the number of Special cars leads to a better and faster distribution of information (1, 2) which by itself should lead to an increase in the routing performance. But on the other hand, when there are many gossip cars that have updated information on the traffic load, they all try to pass through the same empty roads and by doing so make these roads crowded- a fact that leads to a decrease in the routing performance (test 12).

This perception is not correct for other types of networks, for example centralized networks. In centralized networks the availability of information is not related to the number of Special cars and therefore the increase in the number of Special cars leads to a decrease in the routing performance.

Although the gossip model doesn't require any infrastructure in order to distribute the information, its results in the optimum configuration are close to the results in the centralized model (test 12, 13).

The tests show that the optimum number of Special cars is 20%. The tests also give an interesting explanation as to why the optimum is achieved contrarily at 20%. From the measurements checking the characteristics of gossip networks, you can see that at 20% the update rate is faster than the time it takes to traverse an average road (test 1, 2), and the learning probability for a road on average is at least 25% (test 5, 6). These values indicate that the information distribution is very high which leads to the conclusion that it is really not that efficient to increase the percent of gossip cars above 20%.

Another important observation is that the information required for improving the routing performance is very little in relation to all the information distributed. This leads to the conclusion that if a clever way to identify the relevant information is found, it will be sufficient to distribute only that small amount of information. This is

especially important due to the fact that the cars may be driving quickly, and will be able to exchange only a small portion of the information in their databases.

A ranking algorithm was presented in this thesis as a way to identify the relevant information. The results show (test 14) that by using the ranking algorithm and limiting the amount of information exchanged between the cars to 15 roads only (10% of the total information), the routing results were quite close to those expected when all the information was exchanged.

8. Future work

In this work the transportation network simulations focused on a single day scenario. In such a scenario the cars in the network drive from a source to a destination only once. During that time they try to exploit all the information they receive to optimize their voyage.

Another scenario is the multi iteration scenario. In this scenario the simulation mimics several days in which the cars drive from the same source to the same destination every day. In this scenario there is potential for learning from one day to the next. But the information from the previous day may not always be relevant for the following days. For example, if there was a traffic jam at noon on one day, it doesn't mean that it will occur again the following day. In order to overcome this problem the cars can keep a history which takes into account the information from the past. The proportion between the history and the last information (from the previous day) is determined by the learning factor. The learning factor determines how much weight the information from the last day carries in relation to the previous days. For example, if the learning factor is 50% then the current weight is the average between the weight from the last day and the history. This way if the traffic load is continuous and therefore relevant to the future, it will have a major effect. However, if it was only temporary it will not affect the car's future path.

Such a scenario is partially explored in this thesis. It is used in the warm up stage in order to bring the network to a steady point (see chapter 5.8). Because the goal was to bring the network to a balanced situation where the traffic is divided realistically between the roads, there was no need to complicate the situation and insert traffic incidents at this stage. Because there were no traffic incidents, all the information from one day was relevant to the following day. Under these limitations the multi iteration scenario worked very well. In all the cases there was a convergence process where average network load and the average voyage length improved from iteration to iteration until the steady point was reached.

Still, there is much work to be done in investigating the multi iteration scenario in a network including traffic incidents and finding a learning algorithm that will distinguish between the information relevant to the next day and that which is not.

A second subject that can further be explored is routing in gossip networks. In this thesis the characteristics of gossip networks have been explored, and a few measurements have been developed in order to evaluate the information expansion in such networks. These measurements can be an important tool in improving the routing in gossip networks. For example, one of the measurements was the probability of receiving information on a specific road. Cars can use this information in order to calculate a path based not only on the road's weights but also on the probability to learn new information on the way. In such a scenario a car might intentionally select a longer path in order to collect more information because it assumes that eventually it will pay off.

9. References

- [1] D. Helbing, A. Hennecke, V. Shvetsov, M. Treiber "Micro- and macro-simulation of freeway traffic" *Mathematical and Computer Modelling* Volume 35, Issues 5-6 , March 2002, Pages 517-547
- [2] D. Helbing and M. Treiber, Numerical simulation of macroscopic traffic equations, *Computing in Science and Engineering* 1, 89-99 (1999).
- [3] D. Helbing, A. Hennecke, V. Shvetsov and M. Treiber, Macroscopic traffic simulation based on a gas-kinetic, non-local traffic model, *Transportation Research B* 35, 183-211 (2001).
- [4] M. Treiber and D. Helbing, Macroscopic simulation of widely scattered synchronized traffic states, *Journal of Physics A: Mathematical and General* 32, L17-L23 (1999).
- [5] V. Shvetsov and D. Helbing, *Macroscopic dynamics of multi-lane traffic*, *Physical Review E* 59, 6328-6339 (1999).
- [6] Ghulam H. Bham, Rahim F. Benekohal "A high fidelity traffic simulation model based on automata cellular and car following concepts" *Transportation Research Part C: Emerging Technologies* Volume 12, Issue 1 , February 2004, Pages 1-32
- [7] Yang, Q., 1997. *A Simulation Laboratory for Evaluation of Dynamic Traffic Management Systems*. Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.
- [8] Rakha, H., Van Aerde, M., Bloomberg, L., Huang X., 1998. Construction and Calibration of a Large-Scale Microsimulation Model of the Salt Lake Area. *Transportation Research Record* 1644, TRB, National Research Council, Washington, DC, pp. 93–102.
- [9] Nagel, K., Barrett, C.L., 1997. Using micro-simulation feedback for trip adaptation for realistic traffic in Dallas. *International Journal of Modern Physics C: Physics and Computers* 8 (3), 505–525.
- [10] Krauss, S., 1998. *Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics*. University of Cologne, Germany.
- [11] Nagel, K., 1998. From Particle Hopping Models to Traffic Flow Theory. *Transportation Research Record* 1644, Washington, DC, pp. 1–9.
- [12] Daganzo, C.F., 1994. The cell transmission model: a dynamic representation of

- highway traffic consistent with the hydrodynamic theory. *Transportation Research B* 28B (4), 269–287.
- [13] Nurhan Cetin, Kai Nagel *, Bryan Raney, Andreas Voellmy, "Large-scale multi-agent transportation simulations" *Computer Physics Communications* 147 (2002) 559–564
- [14] C. Gawron, An iterative algorithm to determine the dynamic user equilibrium in a traffic simulation model, *Internat. J. Modern Phys. C* 9 (3) (1998) 393–407.
- [15] P.M. Simon, K. Nagel, Simple queueing model applied to the city of Portland, *Internat. J. Modern Phys. C* 10 (5) (1999) 941–960.
- [16] Cetin and K. Nagel. A large-scale agent-based traffic micro simulation based on queue model. In *Swiss Transport Research Conference, Monte Verita, Switzerland, March 2003*.
- [17] DYNAMIT. Massachusetts Institute of Technology, Cambridge, Massachusetts. See its.mit.edu. Also see dynamictrafficassignment.org.
- [18] DYNASMART. See www.dynasmart.com. Also see dynamictrafficassignment.org.
- [19] K. Nagel and M. Rickert. Parallel implementation of the TANSIMS micro-simulation. *Parallel computing* 27(12):1611-1639, 2001.
- [20] Kan Chen, John C. Miles, "ITS Handbook 2000 -Recommendations from the World Road Association (PIARC)", Artech House, 1999.
- [21] J. Boyan and M. L. Littman. "Packet routing in dynamically changing networks: a reinforcement learning approach". In *Advances in Neural Information Processing Systems*, volume 7, pages 671–678, 1994.
- [22] N. Tao, J. Baxter, and L. Weaver. "A multi-agent, policy gradient approach to network routing". In *Proceedings of the Eighteenth International Conference on Machine Learning*, 553-560, 2001.
- [23] D. H. Wolpert, K. Tumer, and J. Frank. "Using collective intelligence to route internet traffic". In *Advances in Neural Information Processing Systems-11*, pages 952–8, Denver, 1998.
- [24] Leonid Peshkin and Virginia Savova, "Reinforcement Learning for Adaptive Routing", In *proceedings of the International Joint Conference on Neural Networks*, 2002.
- [25] Baher Abdulhai 1, Rob Pringle and Grigoris J. Karakoulas, "Reinforcement

- Learning for True Adaptive Traffic Signal Control" *Journal of Transportation Engineering*. 278-285. May 2003
- [26] Fu, L. and Rilett, L.R.(1996) Expected shortest paths in dynamic and stochastic traffic networks, *Transportation Research, Part B: Methodological*, vol. 32, no. 7, pp. 499-516
- [27] Chabini. Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time. *Transportation Research Records*, 1645:170--175, 1998
- [28] C.-L. Liu, T.-W. Pai, C.-T. Chang, and C.-M. Hsieh. Path-planning algorithms for public transportation systems. In *Proceedings of the Fourth International IEEE Conference on Intelligent Transportation Systems*, 1061--1066, 2001
- [29] H. Kanoh, B. Lien, A. Kashiwazaki, et al., "Real-time Route Selection Using Genetic Algorithms for Car Navigation Systems", *IEEE International Conference on Intelligent Vehicles (IV'98)*, pp. 207-212, 1998.
- [30] Bryan Raney, Kai Nagel," An Agent-Based Simulation Model of Swiss Travel" *STRC 03 Conference paper, Monte Verità/ Ascona, March 19 -21, 2003*
- [31] Fertin G. " A study of minimum gossip graphs", *Discrete Mathematics*, 28 March 2000, vol. 215, no. 1, pp. 33-57(25)
- [32] R. Chandra, V. Ramasubramanian, and K. Birman. Anonymous gossip: Improving multicast reliability in mobile ad-hoc networks. In *Proc. 21st International Conference on Distributed Computing Systems (ICDCS)*, pages 275–283, 2001.
- [33] Xiang-Yang Li_ Kousha Moaveninejad Ophir Frieder "Regional Gossip Routing for Wireless Ad Hoc Networks" *Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN'03) 2003 IEEE*
- [34] A. Ebner and H. Rohling, "A self-organized radio network for automotive applications," in *Conference Proceedings ITS 2001, 8th World Congress on Intelligent Transportation Systems, Sydney, Australia, October 2001*.
- [35] Shavitt and Shay, "Optimal Routing in Gossip Networks", 2003 (Unpublished)
- [36] A. Ebner and H. Rohling, "A self-organized radio network for automotive applications," in *Conference Proceedings ITS 2001, 8th World Congress on Intelligent Transportation Systems, Sydney, Australia, October 2001*.
- [37] R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D. S. J. De Couto, "CarNet: A scalable adhoc wireless network system," in *the 9th ACM SIGOPS European*

workshop: Beyond the PC: New Challenges for the Operating System, Kolding, Denmark, Sept. 2000.

- [38] *Wie B.-W.; Tobin R.L.; Carey M*, "The existence, uniqueness and computation of an arc-based dynamic network user equilibrium formulation" *Transportation Research Part B: Methodological*, December 2002, vol. 36, iss. 10, pp. 897-918(22).
- [39] *T.Friesz, D.Bernstein, Z.Suo, R.Tobin* " Dynamic Network User Equilibrium with State-Dependent Time Lags" *Networks and Spatial Economics* (3-4): 319-347, September 2001
- [40] *Yannis A. Korilis, Aurel A. Lazar, Ariel Orda*, "Achieving Network Optima Using Stackelberg Routing Strategies", *IEEE\ACM Transactions on Networking* 1997.