Optimal Routing in Gossip Networks

Yuval Shavitt and Amir Shay

***Abstract*— In this paper we introduce the *Gossip Network* model where travelers can obtain information about the state of dynamic networks by gossiping with peer travelers using ad-hoc communication. Travelers then use the gossip information to recourse their path and find the shortest path to destination. We study optimal routing in stochastic, time independent gossip networks, and demonstrate that an optimal routing policy may direct travelers to make detours to gather information. A dynamic programming equation that produces the optimal policy for routing in gossip networks is presented. In general the dynamic programming algorithm is intractable; however for two special cases a polynomial optimal solution is presented.**

We show that ordinarily gossiping helps travelers decrease their expected path cost. However, in some scenarios, depending on the network parameters, gossiping could increase the expected path cost. The parameters that determine the effect of gossiping on the path costs are identified and their influence is analyzed. This dependency is fairly complex and was confirmed numerically on grid networks.

I. Introduction

Optimal routing in both deterministic and stochastic networks has been extensively studied in the past. While the solutions for the deterministic problem are well known [1] and based on the dynamic programming (Bellman-Ford) or label correcting (Dijkstra) algorithms, the solution to the stochastic problem depends profoundly on the problem modelling. One of the main characteristic of the stochastic problem model is how the information about the stochastic states of the network is obtained. The introduction of ad-hoc communication presents an opportunity for a new kind of network model – the *Gossip Networks*. In this paper we formulate, for the first time, the gossip networks model in which mobile agents obtain information about the state of a stochastic network by exchanging information with neighboring agents using peer to peer (P2P), ad-hoc communication. Mobile agents then use the exchanged information to reveal information about the network state and consequently optimize their routing.

There are varieties of real life problems that can benefit from an optimal solution to the problem of routing in gossip networks. For example, airplanes or vessels that optimize their route by exchanging information with their peers. This paper will focus on another example from the field of transportation. Road congestion is a known and acute urban menace with no signs of disappearing. There are apparently many suggested approaches to tackle this problem; one of them is to supply vehicles and drivers with up-to-date information about road conditions.

There are two kinds of approaches to supply drivers with information that can aid them avoid congestion. One approach is based on fixed-structure communication networks, for example cellular networks or FM/AM radio [2]–[4], the other approach is based on ad-hoc communication networks. Several innovative projects propose using ad-hoc networks as the communication infrastructure, for example FleetNet [5], and CarNet [6].

The advance in technology in recent years helps to bring into vehicle's sophisticated onboard navigation systems at a reasonable price. Such a system contains a computing device with a detailed road map, GPS for locating the vehicle on the map, and communication means. One can use ad-hoc communication networks (such as Wi-Fi) to exchange information between neighboring vehicles. When two vehicles are at communication range they can exchange their information regarding road condition. The road condition information is thus propagated in the network without any need for external or central infrastructure. Each time new information is obtained by a vehicle, the onboard navigation systems recalculate the optimal route from its current location to the destination. For example, if the navigation system receives information that one of the streets in its planned path is blocked it will plan a new path that avoids the blocked road; the new path will be the shortest path from the vehicle's current position to the destination taking into account the blockage.

Our gossip network model was built based on research done in "ad-hoc networks" and "stochastic shortest path routing". In this paper, mobile agents acquire and disseminate information about road conditions using wireless communication (ad-hoc networks) and use the information to minimize their traveling time (shortest path problem). There are two networks in our model, the "road network" on which the mobile agents roam and the "communication network" on which information flow.

While there is an extensive literature about routing in each of the networks, to the best of our knowledge, this is the first attempt to formulate and solve the combined problem: shortest path routing of mobile agents in the context of gossip ad-hoc networks (see also section II-C). [1]

There are currently several ongoing projects focusing on the idea of mobile agents (for example vehicles) exchanging information and forming communication networks without or with a little help from external infrastructure. Mobile Ad-hoc Networks (MANET) [7] is an IETF working group set to standardize these efforts. The FleetNet project [5] aims at the development and demonstration of a wireless ad-hoc network for inter-vehicle communications. FleetNet is a consortium of six companies and three universities looking into mostly the practical issues of providing drivers and passengers some services over ad-hoc communication. Some of the proposed FleetNet services are: notifications about traffic jams and accident, and providing information about nearby available point of interest. Another project, CarNet [6] demonstrates the use of ad-hoc scalable routing protocol (Grid) to support IP connectivity as well as providing services similar to FleetNet. For a comprehensive overview of Inter-Vehicle ad-hoc communication see [8].

FleetNet, CarNet, and similar projects aim at building communication infrastructure using ad-hoc communication and are examining suitable routing protocols; medium access methods, radio modulation, etc. In this paper we assume the existence of such an ad-hoc network that enables mobile agents to exchange information. However, we don't implicitly include here specification of the ad-hoc network such as routing or multi-access communication protocols, instead we abstract them into the *gossip probability*, the probability that a mobile agent will receive information about the status of some roads in the network from another mobile agents. The gossip probability is defined formally in Section II.

The problem of *Shortest Path Routing* was investigated extensively in the literature, for a comprehensive summary of the various efforts in the field of transportation see [9]. In this paper we assume time independence, i.e., the network doesn't change during the course of the travel. Some of the road conditions are known to be alternating, however, a traveler may not know in advance the current condition of all these roads, termed stochastic roads. We assume that no parking at roads or junctions is allowed to optimize the journey, and once a junction is reached the weights of all the roads that emerge from that junction become known. We investigate two different models of weight correlation. The first is the *Independent Weight Correlation* model (G-IWC) where there is no correlation between the states of different edges. The second is the *Dependent Weight Correlation* model (G-DWC) where the network can be in several different states, each state determines the weights of all stochastic edges [10]. Note that the G-IWC model is a generalization of the G-DWC model with substantially more states. The rational behind the G-DWC model is that in "real-life" transportation systems there is a correlation between roads weights, usually a traffic jam in one road effects the roads in its vicinity.

When the shortest path model is stochastic, like in this paper, the information about the actual state of the stochastic edges plays a crucial role in finding the optimal routing solution. Further more, due to the dynamic nature of the problem the solution is not a path but rather a policy that direct the traveler according to the information he obtains. In the literature there are several papers that discuss optimal routing policies in stochastic networks where the traveler can recourse his path according to information obtained during travel. However, the basic difference between these models and ours is that in gossip networks the information is obtained by gossiping with neighboring travelers thus a traveler can obtain data about the state of remote stochastic roads. In all the other models we survey the only way to obtain information about the state of a road is to visit the junction it emanates from. Andreatta and Romeo [11] assume that once a blockage is encountered a recourse path that consists of only deterministic roads is used. Orda, Rom, and Sidi [12] investigated a model where link delay changes according to Markov chains, they model several problems and showed that in general, the problems are intractable. Polychronopoulos and Tsitsiklis [10] investigated a network where there is a correlation between the roads weights. In their model a traveler can deduce the stochastic state by visiting enough roads. Waller and Ziliaskopoulos [13] solved a model with dependency between successor roads and a model with time dependency for the same road.

The primary contribution of this paper is in the introduction and analysis of the gossip model and the new directions it opens for building P2P mobile systems. We choose to introduce the subject using simplified model that allowed us in depth analysis. The analysis presented in this paper produced some interesting results which gives us insight into the characteristics of traveling

[1]This paper focuses on the routing of mobile agents on the "roads networks" and not on the routing of data packets on the "communication networks".

in gossip networks. The introduction of information exchange leads to unique optimal routing policies. In this paper we will show that sometimes it is worth taking a detour to obtain more information about the state of the stochastic edges. The extra cost of the short detour can be compensated by the additional information gained, information that can improve the selection of the continuing path. Further more, we were able to quantify an optimal policy that balance between information gathering costs and path costs. Other main contribution is the regime state diagram we produced. Using the diagram one can determines the influence of gossiping on the traveling costs in different network characteristics.

The rest of the paper in organized as follows. In the next section, the formal model of the gossip networks is introduced and an example that demonstrates the characteristics of the model is presented. An algorithm for optimal routing in gossip networks that is based on dynamic programming is developed in Section III. In Section IV we discuss the implications of traveling in gossip networks. Then, in Section V, we use numerical analysis to demonstrate the influence of the various model parameters on the network behaviors. Finally in Section VI we summarize and highlight our main findings while providing directions for future work.

II. Model and Definitions

A. The formal model

The above discussion leads to the following formal model. The network[2] is represented by a directed graph $G = (V, E)$, where V is the set of vertices, and E is the set of edges, $|V| = n$ and $|E| = m$. An edge $e \in E$ is associated with a discrete random weight variable, w_e. Edges with degenerated weight function that has only one value are termed deterministic, and we denote the set of these edges by $D \subseteq E$. The number of edges in the network with stochastic weights (namely, non deterministic) is denoted by $\delta = |E \setminus D|$. We assume that under all weight distributions there are no negative cost cycles in the network and there is always a path between source and destination.

In the G-IWC model the weights, w_e, of the *stochastic edges* are random variables with discrete probability distribution that has β_e states. The expected cost of an edge is $\bar{w_e} = \sum_{s=1}^{\beta_e} w_e^s q_e^s$, where q_e^s is the probability of an edge e to have the weight w_e^s. We denote by $\hat{w}_e$ the actual weight of the edge e. In the G-DWC model the network can be in only R realizations, each $r \in R$ realization determines the states of the network and thus the weights w_e^r of all the stochastic edges.

Traveling agents (TAs) are roaming the network. Each TA stores internally the weights of the stochastic edges in an *Information Vector*, $I\{\cdot\}$. For example, an information vector of a traveler could look like this: $I = \{\hat{w_1}, X, \hat{w_3}, X, \dots, X, X, \hat{w_\delta}\}$. For known edges, those that the traveler visited or received information about, the weights are written down explicitly, $\hat{w_1}, \hat{w_3}, \hat{w_\delta}$. Unknown edge weights are denoted by X. The number of possible states of the information vector in the G-IWC model, l_I is given by

$$l_I = \prod_{e \in E \setminus D} (\beta_e + 1) \tag{1}$$

and in the G-DWC model, the number of different information vector states is given by

$$l_D = \sum_{i=1}^{R} \binom{R}{i} = 2^R - 1 \tag{2}$$

When two or more TAs are within communication range they can exchange their information vectors in order to gain missing data. The *gossip probability* is the probability that when a TA traverses an edge it will update his information vector.

$$P(s, s', T(i,j)) = \mathcal{P}\{I(j) = s' | I(i) = s, T(i,j)\} \tag{3}$$

where $s, s' \in I$ are the information vector before and after the edge (i,j) traversal, respectively, $I(i)$ is the information vector at vertex $i \in V$, and $T(i,j)$ is the *topology probability*. The topology probability is the probability that a TA will receive information from other TAs during the traversal on an edge. The topology probability is determined by aspects like the number of TAs around the traveler, the other TAs previous paths, physical obstacles that interfere with the wireless communication, etc. It is a characteristic of the network structure and the flows of TAs in the network. Assuming that there are ”enough“ mobile agents in the network $T(i,j)$ is a vector of probabilities, where each element corresponds to some stochastic network edge. For example, $T(i,j) = \{1, 0.5, \dots 0\}$ means that on average when the TA slates edge (i,j) it will learn about stochastic edges 1,2, and δ with probability 1, 0.5, and zero, respectively. The gossip probability depends on the topology probability and on the information vector before and after the edge traversal. For example, the probability to change an information vector element from

[2]As mentioned above, there are two networks in our model, the “road network” and the “communication network”. In this paper, when we say “network” we refer to the “road network”. We assume the existence of communication network that enables mobile agent to exchange information but in this paper we don’t include it in the formal model implicitly, it is included in the gossip probability presented below.

$\{\cdots, \hat{w}, \cdots\}$ to $\{\cdots, X, \cdots\}$ is zero. Regardless of the topology probability, a known weight can not be changed into unknown.

In this paper we are looking for the optimal routing policy of a TA that start at the source vertex s with information vector $I(s)$ and travels to a destination vertex t. We assume that the TA knows a priori the network structure, weights distribution, and the topology probability. We are looking for an optimal routing policy, π^* with minimal expected cost, $C^*(s,t,I(s))$, of all possible routing policies $\pi^k \in \pi$.

$$\forall\ \pi^k \in\ \pi \quad C^*(s,t,I(s)) \leq C^k(s,t,I(s))$$

B. Assumptions and Reality

The formal model of this paper has several assumptions. In this section we summarize these assumptions and relate them to real life scenarios in transportation networks. The first assumption is that the network is time independent. In many situations, a driver can assume that during his commute (30 to 60 minutes) the traffic patterns in his area doesn't change significantly. Thus, in many cases, an optimal routing policy calculated at the beginning of the journey will yield satisfying results throughout the journey.

Another assumption is that the agent knows a priori the network structure, edges weight distribution and topology probability. While network structure can be obtained from any GIS, the edges weight and topology probability are calculated from historical information gathered over time. Currently there are several commercial and academic projects that use historical data to predict future traffic patterns, for example the MIT's DynaMIT project [14]. While the edges weight distribution can be computed directly from the historical traffic data, in order to compute the topology probability one needs information about the agents movement in the network. Given that information, we can calculate and record fairly easily the edges weight distribution and topological probability for a given time. For example, we will have one distribution for morning commute, second for evening commute, third for holidays etc. Then, each time the agent will compute his optimal routing policy using the gossip network time independent algorithm he will use the appropriate distributions.

Any probability distribution is meaningful only when there are enough events. Thus, in order to calculate the edges weight distribution one needs ”enough“ historical information both over time and network edges. The calculation of the topology probability requires information from ”enough“ agents in the network. In a study done by Kraus, Parshani, and Shavitt [15] it was shown that when the portion of cars with gossiping capability is anywhere between 1% to 60% there is an reduction in the average delay of the gossiping cars, and in most of the region also in the average delay of the total car population. Researchers at DLR were able to deduct meaningful information about real time traffic using several hundred taxis in Berlin, Nuremberg, and Vienna [16].

C. Comparison with other Ad-Hoc Models

There is a fairly large body of work that deals with gossiping in ad-hoc networks, however the model and thus techniques used in these works is different from our work. In general the goal in most of the ad-hoc network literature is to seek efficient protocols for information exchange minimizing communication overhead, power consumption etc, while ensuring message delivery. The main focus of our paper is to propose an optimal routing algorithm that minimize travel costs.

In this paper gossiping is used to exchange information about the weights of the stochastic edges between agents. Hass, Halpern, and Li [17] build ad-hoc routing protocols where gossiping is used to reduce the protocols overhead. Kulik, Rabiner, and Balakrishnan [18] proposed the SPIN family of protocols that use gossiping to overcome problems such as implosion, overlap, and resource blindness common to ad-hoc networks. Braginsky and Estrin [19] introduced a scheme that allow queries to be delivered while providing tradeoff between setup overhead and delivery reliability. While it is possible to combine our modelling and results for these works it is certainly not straight forward due to the differences between the underlaying assumptions.

First, in our model, the network topology is assumed to be known to a large extent and mostly the weights are unknown. In ad-hoc networks, the network is assumed to change so frequently that the overhead to learn its topology is too large to become realistic. Thus, in our case we collect knowledge about the state of edges, while in ad-hoc networks the effort is to learn a route (sometimes with the ability to improve it based on cost) but there is no attempt to learn the network state and optimize based on this.

Another major difference, is that in our case we assume the existence of a priory knowledge about the statistics of the network, such as, the weight distribution of the links, the probability to learn about the state of a certain link by traveling on another, etc. In most other ad-hoc network models, such knowledge is never assumed.

In other ad-hoc networks, one has a full control on the ability to distribute information about the networks by changing the control algorithm. In our case, information

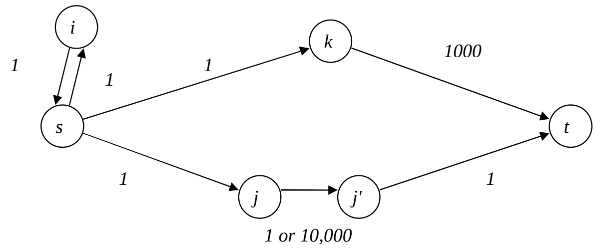

Fig. 1. An example of the influence of gossiping on routing. We are looking for the optimal routing policy between the vertices s and t where the edge (j, j') is stochastic and on edge (i, s) the traveler can obtain information about the stochastic edge. The path {s,i,s} is called an 'information gathering loop' (IGL)

is flooded by cars whose drivers selected to mount special gossip equipment, but the drivers are going on their own private business. Thus we do not control the rate and direction of the information dissemination. This lack of control disqualify many of the solutions suggested in the context of ad-hoc networks in our model.

D. An Example

In the example network presented in Fig. 1, a traveler is located at vertex s and is looking for the optimal routing policy to vertex t. In this network there is one ($\delta = 1$) stochastic edge, (j, j'), that has two possible states. With probability $q^u_{jj'} = \xi_U$ the edge is in the "UP" state where $w^u_{(j,j')} = 1$, and with probability $q^d_{jj'} = (1 - \xi_U)$ the edge is in the "DOWN" state where $w^d_{(j,j')} = 10000$. The traveler can obtain information about the state of the edge (j, j') only when traversing the edge (i, s), with a topology probability of $T(i, s) = \xi_T$. The gossip probability of this network is:

$$
\begin{aligned}
P(\{X\}, \{X\}, T(i, s)) &= 1 - \xi_T \\
P(\{X\}, \{1\}, T(i, s)) &= \xi_T \\
P(\{X\}, \{10000\}, T(i, s)) &= \xi_T \\
P(\{1\}, \{1\}, T(i, s)) &= 1 \\
P(\{10000\}, \{10000\}, T(i, s)) &= 1 \\
\text{Else } \forall\ u, v \in V \quad P(I(u), I(v), T(u, v)) &= 0
\end{aligned}
$$

The traveler has to choose between different travel options: *a*) The "safe" path through vertex k which guarantee a cost of 1001 or; *b*) The "risky"[3] path through vertex j with cost that depends on the state of edge (j, j'), either 10002 or 3 or; *c*) Travel to vertex i, obtain information about the status of edge (j, j') and then, according to the obtained information, choose whether to go through vertex k, j or return to vertex i.

[3]The risky policy is taken by a traveler that must reach the destination at some specific time (for example to catch a plane that leaves in 10 time units). If not there by that time the traveler care less about the path cost (anyway he needs to reschedule).

Next we will calculate the expected cost of the different routing policies. The cost of the path through vertex k is deterministic and does not depend on the a priori knowledge of the state of the edge (j, j')

$$C(s, t, \{\cdot\})_k = 1001 \tag{4}$$

The cost of the path through vertex j without any a priori knowledge about the state of the edge (j, j')

$$C(s, t, \{X\})_j = 10002(1 - \xi_U) + 3\xi_U \tag{5}$$

If the traveler needs to choose between traveling through k or j (without first traveling to vertex i) then his optimal routing policy depends on the value of his information vector:

$$
\begin{aligned}
C^*(s, t, \{X\})_{kj} &= \min(1001, (1 - \xi_U)10002 + 3\xi_U) \\
C^*(s, t, \{1\})_{kj} &= 3 \\
C^*(s, t, \{10000\})_{kj} &= 1001
\end{aligned}
$$

If the traveler knows that the stochastic edge is in the "DOWN" state he will travel to vertex k; in the case he knows that the edge is in the "UP" state he will travel to vertex j; and in the case the traveler doesn't know the state of the stochastic edge he will decide according to the value of ξ_U.

When the traveler moves to vertex i without any a priori knowledge about the state of the edge (j, j') the expected cost of his routing policy assuming one trial to obtain information is:

$$
\begin{aligned}
C(s, t, \{X\})^{(1)}_i &= 2 + \xi_T[\xi_U C^*(s, t, \{1\})_{kj}] + \\
&\quad (1 - \xi_U)C^*(s, t, \{10000\})_{kj}] + \\
&\quad (1 - \xi_T)C^*(s, t, \{X\})_{kj} \\
&= 2 + \xi_T[3\xi_U + 1001(1 - \xi_U)] + \\
&\quad (1 - \xi_T)C^*(s, t, \{X\})_{kj}
\end{aligned} \tag{6}
$$

When the traveler routing policy is to cycle between vertices s and i until he obtains information, the expected number of cycles he will need is $1/\xi_T$. Therefore

$$C(s, t, \{X\})_i = 2(1/\xi_T) + 3\xi_U + 1001(1 - \xi_U)$$

For the above example there is a threshold topological probability, ξ_0, such that for $\xi_T \geq \xi_0$

$$C^*(s, t, \{X\})_i < C^*(s, t, \{X\})_{kj} \tag{7}$$

Meaning that for $\xi_T \geq \xi_0$ the traveler's optimal routing policy when there is no information is to make a detour through node i until it obtains information about the state of the stochastic edge. In this paper we call the path $\{s, i, s\}$ an *Information Gathering Loop* (IGL).

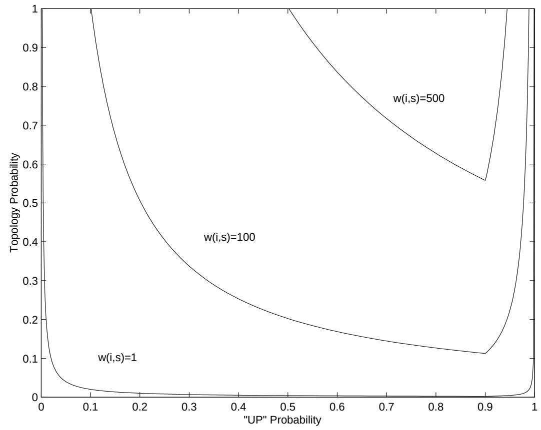

Fig. 2. The relation between the "UP" and gossip probabilities for different $w_{(i,s)}$ values. The area above the line is where $C^*(s,t,\{X\})_i < C^*(s,t,\{X\})_{kj}$ and the traveler will cycle for information

Topological Probability(ξ_T)	$I(s)$	next hop
$\geq \xi_0$	$\{X\}$	i
$\geq \xi_0$	$\{1\}$	j
$\geq \xi_0$	$\{10000\}$	k
$< \xi_0$	$\{X\}$	α
$< \xi_0$	$\{1\}$	j
$< \xi_0$	$\{10000\}$	k

TABLE I
ROUTING TABLE OF THE SOURCE VERTEX s. THE VALUE OF α IS k OR j ACCORDING TO THE VALUE OF ξ_U.

Fig. 2 illustrates this by plotting the equilibrium line of Eq. (7) for different values of $w_{(\hat{i,s})}$. The area above the line is where the inequality holds and the traveler is making a detour to gather information. The minimum of the plots in Fig. 2 is when Eq. 5 and Eq. 4 are equal, for $w_{(\hat{i,s})}$ at $\xi_U = 0.90028$ in this example.

The optimal routing policy for a traveler that starts on vertex s is outlined in the EXAMPLE_POLICY below. And the corresponding routing table for source vertex s is outlined in Table I.

EXAMPLE_POLICY

IF $\xi_T \geq \xi_0$
  WHILE $I = \{X\}$ cycle in the path $\{s, i, s\}$
IF $I = \{1\}$
  Then take the path $\{s, j, j', t\}$
ELSE IF $I = \{10000\}$
  Then take the path $\{s, k, t\}$
ELSE IF $I = \{X\}$
  Then take the path $\min(\{s, j, j', t\}, \{s, k, t\})$
END

III. The Routing Algorithm

A. Solution approach

The problem of finding the optimal routing in gossip networks belongs to the class of online decisions problems. In these problems an agent is faced with the opportunity of influencing the behaviors of a probabilistic system as it evolve. At each step the agent receives information about the system state and performs an action accordingly. His goal is to choose a sequence of actions which causes the system to perform optimally with respect to some predetermine criteria. Due to the stochastic nature of the system decisions must anticipate the costs associated with future system states. In the literature such problems can be found under the topics of *Markov Decision Processes* [20], stochastic programming [21] and optimal control [22]. Similar to other online decisions problems, we solve the problem of optimal routing in gossip networks using dynamic programming and in general share the same "curse of dimensionality" [23], which lead to intractable solution. What is unique about our model is the way the agents learn about the state of the network. An optimal policy in gossip networks needs to seek the optimized balance between the path cost and the cost of gathering information. For example, the optimal policy might direct the agent to a path with higher cost but with higher probability to gather important information. This policy will reduce the agent's total expected cost. Unlike most of the online decisions problems, in gossip networks decisions must anticipate both the edge costs and the information gathering opportunities associated with future system states. It is well known throughout the online decision problem literature that information pays off, in our algorithm we were able to quantify the importance of information.

The optimal routing policy in gossip networks is the one with the minimum expected cost from source to destination for a given information vector. Next we will show how one can calculate the expected cost of a routing policy in the network, in the next subsection we will introduce an algorithm that uses these calculations to find the optimal routing policy to a destination.

A traveler starts his journey from vertex s with information vector $I(s)$ and wishes to reach vertex t. During his journey, there is a probability that he will learn, through gossiping, about the states of the stochastic edges and accordingly update his information vector $I(\cdot)$. At every vertex $r \in V$ he reaches, the traveler makes a routing decision, based on his updated information vector. The expected cost of a routing policy between a source vertex, s, and a destination vertex, t, through a neighbor vertex, r, is:

$$C(s,t,I(s))_r = \hat{w}_{(s,r)} + \sum_{I(r)\in B(I(s),(s,r))} P(I(s),I(r),T(s,r)) \cdot Q(I(r)) \cdot C(r,t,I(r)) \quad (8)$$

The weight of edge (s,r) is known and its value is $\hat{w}_{(s,r)}$. $B(I(s),(s,r))$ is the set of all the possible information vectors $I(r)$ of the traveler when reaching vertex r, assuming that at vertex s it has the information vector $I(s)$. $P(I(s),I(r),T(r,s))$ is the gossip probability that the information vector will change from $I(s)$ into $I(r)$ on the edge (s,r). $Q(I(r))$ is the a priori probability that the network G is in a state corresponding to the information in $I(r)$.

B. Dynamic Programming Algorithm

In this section we present the GOSSIP_DP algorithm that builds the optimal routing tables for gossip network, the algorithm is outlined in Fig. 3. A formal proof of the algorithm correctness is provided below in section III-D.

The optimal routing policy from vertex s to vertex t in the gossip networks, $C^*(s,t,I(s))$, is the one that minimizes the expression in Eq. 8. Namely, the one that selects the policy with the smallest expected cost. Thus, we can write the following dynamic program:

$$C^*(s,t,I(s)) = \min_{r\in\mathcal{N}_s}\{w^{I(s)}_{(s,r)} + \sum_{I(r)\in B(I(s),(s,r))} P(I(s),I(r),T(s,r)) \cdot Q(I(r)) \cdot C^*(r,t,I(r))\} \quad (9)$$

where $\mathcal{N}_i$ is the group of neighbors of vertex i and $w^{I(s)}_{(s,r)}$ is the weight of the edge (s,r) assuming that the information state before is $I(s)$. When the information vector contains information about the state of vertex (s,r) the weight is known $\hat{w}$, in all other cases we take the weight to be the expected weight $\bar{w}_{(s,r)}$ over all the states according to the value of $I(s)$.

In Bellman-Ford's dynamic programming algorithm for deterministic shortest path [1] one finds for each vertex the shortest path to a destination. In gossip networks, using the algorithm GOSSIP_DP in Fig. 3, we find for each vertex the shortest path for each possible state of the vertex's information vector $I(\cdot)$.

Specifically, for each vertex $u \in V$ we keep a table, $TBL[u]$, (see Fig. 4) that has l rows (l is defined in Eq. 1 or Eq. 2 according of the model in use). Each row holds the information vector state ($s_k \in I$) the distance to destination, (DD) and a pointer to next vertex (PN).

```
Algorithm GOSSIP_DP(G, w, T, s, t)
≪ Initialize the routing tables ≫
for k = 1 to l
    DD[t, s_k] ← 0; PN[t, s_k] ← t
Cont ← true
for each u ∈ V \ t
    for k = 1 to l
        DD[u, s_k] ← ∞; PN[u, s_k] ← NIL
≪ Main Loop ≫
while Cont = true
    Cont ← false
    for each e ∈ E
        if G_RELAX(e) then Cont ← true
end
≪ Relax the entry for the edge ≫
function G_RELAX(e)
    u ← Source(e); v ← Destination(e)
    Relax ← false
    for k = 1 To l
        tempDD ← w_e^{s_k} +
            Σ_{m=1}^{l} T_PRB(s_k, s_m) · DD[v, s_m]
        if DD[u, s_k] − tempDD > ε then
            DD[u, s_k] ← tempDD
            PN[u, s_k] ← v
            Relax ← true
    next k
    return(Relax)
end function
≪ The transition probability s_k → s_m ≫
function T_PRB(s_k, s_m)
    P ← prob. to move from s_k to s_m on e
    Q ← prob. of the network to be in s_m
    return(P · Q)
end function
```

Fig. 3. The GOSSIP_DP algorithm.

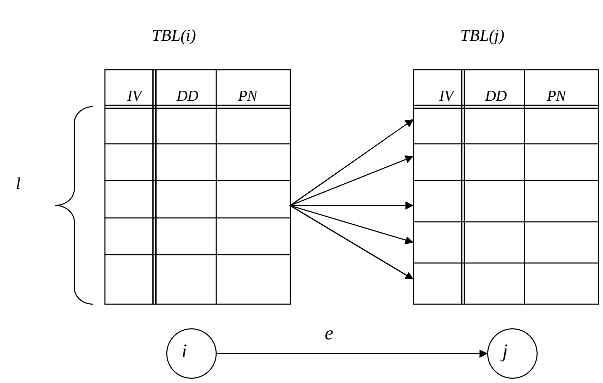

Fig. 4. The relaxation process for one state of one edge.

The first steps of the GOSSIP_DP, lines 1 to 6, initialize this data structure.

In the main loop of the algorithm, lines 7 to 11, we iterate over all the edges of the network and relax each edge. This loop continues while at least one of the edges was relaxed.

In the function G_RELAX we relax for a specific edge all the possible information vectors. The relaxation processes for each edge (u, v) and for each information vector state s_k, lines 16 to 20 is:

$$\begin{aligned} DD[u, s_k] &= w^{s_k}_{(u,v)} + \\ &\sum_{m=1}^{l} P(s_k, s_m, T(u, v))Q(s_m)DD[v, s_m] \end{aligned} \quad (10)$$

For each source vertex state, s_k, we check what is the probability that during the travel on the edge (u, v) the state s_k will change into s_m, $(m = 1 \dots l)$. Each gossip probability $P(s_k, s_m, T(u, v))$ is multiplied by the destination vertex distance $DD[v, s_m]$ and the probability $Q(s_m)$ that the network will be in state s_m.

The iterations stop when for all edges and information vectors the difference between iterations weights is less than ϵ, as shown in line 17. In the classical algorithm $\epsilon = 0$, in our case ϵ is a small positive constant. This condition comes to overcome a situation that our network contains an information gathering loops as we saw in the example section II-D above and illustrates in section III-C below. The parameter ϵ is chosen so that $\epsilon \ll w^{s_k}_{(u,v)}$ for all edges , $(u, v) \in E$, and information states, $s_k \in I(s)$, so that it will come into play only when there are information gathering loops. For a complete discussion of the stoping conditions see the proof of the algorithm correctness in section III-D.

The algorithm GOSSIP_DP is used to produce the optimal routing policy in gossip networks by the following steps: Before the traveler starts his journey he builds his optimal routing policy by calculating $TBL[\cdot]$ for all the vertices of the network using the algorithm GOSSIP_DP. During his journey the traveler updates his information vector and navigates on the network using the information in $TBL[\cdot]$. Every time the traveler reaches a new vertex $u \in V$ with information vector state $s_k = I(u)$ he looks for the next vertex in $PN[u, s_k]$. Later in Section V, we use the GOSSIP_DP to derive our numerical analysis.

C. GOSSIP_DP Execution Example

Next we will explore the behavior of the algorithm GOSSIP_DP on a network with an information gathering loop, like the one presented in Fig. 1. In the following discussion the information gathering loop has two edges, the first with the cost of L_1, the second with the cost of L_2, the total costs of the loop is $L = L_1 + L_2$. When we travel on the second edge of the loop the probability to gather information is $\xi_T = P(\{X\}, \{0/1\}, T(i, s))$. The optimal cost from source s to destination t when the traveler has information $(I(s) \neq \{X\})$ is Z and without information $(I(s) = \{X\})$ is Y. Following the dynamic programming iterations, when vertex s is k hops from the destination the optimal cost is

$$\begin{aligned} DD^k[s, \{X\}] &= Y \\ DD^k[i, \{X\}] &= \infty \end{aligned}$$

The optimal cost from vertex i to destination is infinity due to the fact that for k hops there is no path from i to destination. Moving to the next iteration of the dynamic programming and adding one hop we get for the optimal cost with $k + 1$ hops

$$\begin{aligned} DD^{k+1}[s, \{X\}] &= Y \\ DD^{k+1}[i, \{X\}] &= L_2 + \xi_T \cdot Z + (1 - \xi_T)DD^k[s, \{X\}] \\ &= L_2 + \xi_T \cdot Z + (1 - \xi_T)Y \end{aligned}$$

The costs of $DD^{k+1}[i, \{X\}]$ was calculated using Eq. 10. After adding another hop to the optimal cost

$$\begin{aligned} DD^{k+2}[s, \{X\}] &= L_1 + DD^{k+1}[i, \{X\}] \\ &= L + \xi_T \cdot Z + (1 - \xi_T)Y \\ DD^{k+2}[i, \{X\}] &= L_2 + \xi_T \cdot Z + (1 - \xi_T)Y \end{aligned}$$

In the $(k + 2)$ iteration the dynamic programming choose to cycle in the loop instead of traveling directly to destination. For that to happened the expected cost of the path with a loop should be smaller than the path without a loop, and mathematically

$$\begin{aligned} DD^{k+2}[s, \{X\}] &< Y \\ L + \xi_T \cdot Z + (1 - \xi_T)Y &< Y \\ L &< \xi_T(Y - Z) \end{aligned} \quad (11)$$

The weight of the loop (L) should be smaller than the costs of expected gain from the information in the loop $(\xi_T(Y - Z))$. After adding another hop we receive

$$\begin{aligned} DD^{k+3}[s, \{X\}] &= L_1 + DD^{k+2}[i, \{X\}] \\ &= L + \xi_T \cdot Z + (1 - \xi_T)Y \\ DD^{k+3}[i, \{X\}] &= L_2 + \xi_T \cdot Z + (1 - \xi_T)DD^{k+2}[s, \{X\}] \\ &= L_2 + \xi_T \cdot Z \\ &+ (1 - \xi_T)(L + \xi_T \cdot Z + (1 - \xi_T)Y) \end{aligned}$$

In the general case, for a path with $k+2n+1$ hops we receive

$$\begin{aligned} DD^{k+2n}[s,\{X\}] &= L_1 + DD^{k+2n}[i,\{X\}] \\ &= L + \xi_T \cdot Z \\ &+ (1-\xi_T)[L + \xi_T \cdot Z \\ &+ (1-\xi_T)^2(L + \xi_T \cdot Z) \\ &+ \cdots + (1-\xi_T)^{n-1}(L + \xi_T \cdot Z) \\ &+ (1-\xi_T)^n Y] \\ &= (L + \xi_T \cdot Z)\sum_{j=0}^{n-1}(1-\xi_T)^j \\ &+ (1-\xi_T)^n Y \\ &= (L + \xi_T \cdot Z)((1-(1-\xi_T)^{n-1})/\xi_T \\ &+ (1-\xi_T)^n Y \end{aligned}$$

For each two hops we add in the dynamic programming the optimal path adds another cycle. The endless cycling is due to the fact that each cycle reduce the optimal cost. However, the costs of the optimal policy with endless cycling converge,

$$\lim_{n\to\infty} DD^{k+2n}[s,\{X\}] = L/\xi_T + Z \quad (12)$$

One should notice that although the optimal policy in this case instruct the traveler to cycle endlessly when he has no information about the network state, with probability one the traveler will not cycle endlessly. When the traveler follows this policy he will eventually gather information and then uses suitable policy to the destination.

In summary, when the optimal policy has cycles, following the condition in Eq. 11, consecutive iterations of the dynamic programming continues to instruct the optimal path to cycle, where each iteration decreases the optimal policy costs, this value converge to $L/\xi_T + Z$ in our example.

If we choose some ϵ and stop the dynamic programming when the cost improvement between consecutive iterations is smaller than ϵ, in our example when

$$0 \le DD^{k+j}[s,\{X\}] - DD^{k+j+2}[s,\{X\}] < \epsilon$$

then we are certain that the dynamic programming algorithm stops after a finite number of steps with a policy which is optimal or at most ϵ away from optimal. A formal proof is given in the next section.

D. GOSSIP_DP Correctness

The proof that the algorithm GOSSIP_DP in Fig. 3 provides the optimal solution for routing in gossip networks is a direct extension of a deterministic Bellman-Ford proof [1]. There are two main differences between our algorithm and the classical one. The first difference lays in the fact that in GOSSIP_DP there are several of possible information states for each vertex compared to one deterministic state in the classical Bellman-Ford algorithm. Another major difference lays in the fact that in GOSSIP_DP network loops can turn to be beneficial as illustrated in section III-C.

Consider the GOSSIP_DP algorithm in Fig. 3 and assume the following:
(i) There is at least one path from each vertex $v \in V$ and state $s_k \in I(\cdot)$ to destination t.
(ii) There are no negative weight cycles in the graph G.

Denote by $TBL^i[v,s_k]$ the routing tables of vertex v with information vector s_k when the length of the path from the source vertex v to the destination vertex t has at most i hops. The relaxation presented in Eq. 10 can be written as

$$DD^{i+1}[v,s_k] = \min_u [w^{s_k}_{(v,u)} + \bar{DD}^{i,s_k}_{(vu)}]$$

Where we used the initialization $\forall\ i\ ,\ \forall\ s_k \in I(\cdot)\ \ DD^i[t,s_k] = 0$ and $\bar{DD}^{i,s_k}_{(vu)}$ is the expected weight over all the possible information vector states

$$\bar{DD}^{i,s_k}_{(vu)} \equiv \sum_{s_j=1}^{l} P(s_k, s_j, T(v,u))Q(s_j)DD^i[u,s_j] \quad (13)$$

In the following we define an *iteration* as preforming the relaxation process for all the possible edges $e \in E$ and for each edge for all its possible information states, $s_k \in I(\cdot)$.

We begin our algorithm correctness proof with three lemmas. The first, Lemma 3.1, proves that in each iteration the algorithm's routing tables contain the optimal policy. The second, Lemma 3.2, and the third, Lemma 3.3, prove that the algorithm terminates with the optimal polices. The second lemma (3.2) in for the case of a network without information gathering loops and the third lemma (3.3) with them.

Lemma 3.1 (GOSSIP_DP Optimal Policy): The values of the routing tables $TBL^i[v,s_k]$ generated by the GOSSIP_DP algorithm contain the optimal policy information for v, s_k and i.

Proof:

We prove by induction on the maximum number of hops in a policy path.

For the induction base, we observe that the routing tables for paths with a length of one edge is

$$DD^1[v,s_k] = w^{s_k}_{(v,t)} \quad \forall\ \ v \in V, s_k \in I$$

For all vertex $u \in V$ that are not neighbors of the destination t we denote, $w^{s_k}_{(u,t)} = \infty$. So $DD^1[v,s_k]$ is indeed equal to the optimal policy from v to t for paths with length ≤ 1.

Suppose that $TBL^i[v, s_k]$ contain the optimal policy with paths that contain at most i hops from all $v \in V$ and for all $s_k \in I$. We will now show that $TBL^{i+1}[v, s_k]$, we construct in the GOSSIP_DP algorithm, contain the optimal policy for paths that contain at most $i + 1$ hops from all $v \in V$ and for all $s_k \in I$. Indeed, an optimal policy from v to t either consists of less than $i+1$ hops, in this case $TBL^i[v, s_k]$ contains the optimal policy, or else it consists of $i+1$ hops with the first being (v, u) for some u, followed by an i-edge policy from u to t. The latter policy must be the optimal policy to reach t from u with a length shorter than $i + 1$ hops (otherwise we could use the optimal policy with at most i and obtain a better policy for at most $i + 1$). Denoting the cost of the optimal policy that contains at most $i + 1$ hops by OP^{i+1}

$$OP^{i+1} = \min\{DD^i[v, s_k], \min_u(w^{s_k}_{(v,u)} + \bar{DD}^{i,s_k}_{(vu)})\} \quad (14)$$

Using the induction hypopiesis, we have $DD^m[v, s_k] \leq DD^{m-1}[v, s_k]$ for all $m \leq i$. The set of policies that has at maximum m hops contains the corresponding set of polices that has at maximum $m - 1$ hops. Therefore

$$\begin{aligned} DD^{i+1}[v, s_k] &= \min_u[w^{s_k}_{(v,u)} + \bar{DD}^{i,s_k}_{(vu)}] \quad (15) \\ &\leq \min_u[w^{s_k}_{(v,u)} + \bar{DD}^{i-1,s_k}_{(vu)}] \\ &= DD^i[v, s_k] \end{aligned}$$

Furthermore, we have for all $v \in V$ and $s_k \in I$

$$DD^i[v, s_k] \leq DD^1[v, s_k] = w^{s_k}_{(v,t)} = w^{s_k}_{(v,u)} + DD^i[t, s_k]$$

Thus from Eq. 14 we obtain

$$\begin{aligned} OP^{i+1}[v, s_k] &= \min\{DD^i[v, s_k], \min_u(w^{s_k}_{(v,u)} + \bar{DD}^{i,s_k}_{(vu)})\} \\ &= \min\{DD^i[v, s_k], DD^{i+1}[v, s_k]\} \end{aligned}$$

In view of Eq. 15, $DD^{i+1}[v, s_k] \leq DD^i[v, s_k]$, this yields

$$OP^{i+1}[v, s_k] = DD^{i+1}[v, s_k]$$

Completing the induction proof. ■

Lemma 3.2 (GOSSIP_DP Termination Without IGL): The algorithm GOSSIP_DP terminates after $j < |V|$ iterations, when there are **no** information gathering loops in the network. At termination $PN^j[v, s_k]$ contains the optimal policies.

Proof:

In Lemma 3.1 we proved that at any iteration the routing tables contain the optimal policy for that iteration. Here we need to prove that the algorithm terminates, and that it doesn't terminate too soon - running the algorithm further will not reduce the optimal cost.

For a given information state adding hops to the optimal policy could reduces its cost, until the optimal policy contains at most all the edges of the network. Adding more hops in this situation can only increases the policy costs under the assumption that there are no negative weight loops in the network. Thus at some point the termination condition, line 17 of the algorithm GOSSIP_DP, will come into effect and terminates the algorithm. In the iterations notation this condition becomes

$$\begin{aligned} &\forall\ v \in V, s_k \in I(\cdot) \\ &\quad 0 \leq DD^{h-1}[v, s_k] - DD^h[v, s_k] < \epsilon \end{aligned} \quad (16)$$

The algorithm terminates with the optimal policies and not before due to the fact that the stoping condition in Eq. 16 does not come into effect until the optimal path contains the optimal number of hops.

The value of ϵ is chosen such that

$$\forall\ \ (u, v) \in E, s_k \in I(\cdot)\ \ \epsilon \ll w^{s_k}_{(u,v)}$$

In each iteration until all optimal paths are found, at least one vertex decreases its current cost in the order of an edge weight.

Thus when there are no IGL in the network the algorithm GOSSIP_DP terminates after at most $|V|$ iterations and when it terminates the routing tables contain the optimal polices.

■

Lemma 3.3 (GOSSIP_DP Termination With IGL): The algorithm GOSSIP_DP terminates after at most $j = f(\epsilon)$ iterations, when there **are** information gathering loops in the network. At termination $PN^j[v, s_k]$ contains the optimal policies up to a factor of ϵ.

Proof:

Following Lemma 3.2 here we need to demonstrate the effect of adding IGLs to the network. We illustrated in section III-C that when adding an IGL to a network at some point the optimal policy directs the traveller to cycle. Each cycle reduces the policy costs further due to the increase in the probability to gather information. Thus if the optimal policy starts to cycle, it will cycle forever. The stoping condition, Eq. 16, ensures that the algorithm stops and doesn't run forever. Because the optimal policy is set to cycle forever, stoping the cycling under the conditions in Eq. 16 doesn't change the optimal policy. However, we stop the cycling and do not allow the optimal policy costs to converge to its final value. Thus the loop can carry an error in the order of ϵ. At most we can have $m = |E|$ loops in the network thus the overall error is $O(m \cdot \epsilon)$. If we define $\epsilon' = \epsilon \cdot m$, we can conclude that at termination $PN^j[v, s_k]$ contains the optimal policies up to a factor of ϵ'. ■

Theorem 3.1 (GOSSIP_DP Correctness): The algorithm GOSSIP_DP provides the optimal policy for gossip networks when there are no information gathering loops (IGLs). When there are IGLs the algorithm provides an optimal $+\epsilon$ approximation.

Proof:

In order to show the GOSSIP_DP algorithm correctness we need to prove the following

(a) At each iteration the algorithm contains the optimal policy for that iteration. This was proved in Lemma 3.1.

(b) When the network **doesn't** contain information gathering loop the algorithm terminates with the optimal policy after $j < |V|$ iterations. This was proved in Lemma 3.2.

(c) When the network **contain** information gathering loop the algorithm terminates with the optimal policy up to a factor of ϵ after $j = f(\epsilon)$ iterations. This was proved in Lemma 3.3.

■

E. Complexity of G-IWC and G-DWC

Theorem 3.2: In the case there are no information gathering loops in the network the complexity of the GOSSIP_DP algorithm under the G-IWC model is $O(nm\delta(2\beta+1)^{\delta})$.

Proof: When there is no correlation between the edges weights we must examine all the edges ($O(|E|)$); for each edge we must examine all the source vertex stochastic states ($O(l_I)$); and for each source vertex stochastic state we examine all the destination vertex's stochastic state ($O(l_I)$), here we assume that the number of stochastic states is bounded by β. Notice however that not all state transfers are possible and actually the number of possible state transfer we need to examine is only $(2\beta+1)^{\delta}$. The first $\beta+1$ states are for the transfer from state $\{X\}$ to all the available states, the second β states are for staying in the same state when the weight of the stochastic edge is known. In each state transfer we need to calculate the transfer probability P and the a priori probability Q, for that we need to examine all stochastic edges $O(\delta)$. In the worst case a vertex has $O(|V|)$ neighbors and the algorithm terminates either after repeating for each of the neighbors or when there is no difference between successive iterations. ■

Theorem 3.3: The complexity of the GOSSIP_DP algorithm under the G-DWC model is $O(nm\delta 2^{2R})$.

Proof: The complexity of GOSSIP_DP algorithm under the G-DWC model is similar to the complexity of the algorithm under the G-IWC model. The only different is that we need to examine $O(l_D)$ transfer states instead of $O(l_I)$ states. According to Eq. 2, $O(l_I) = 2^R$. ■

Although the optimal solution to the gossip networks problem is intractable in general, we presented above two special cases where the optimal solution is polynomial in respect to the network size. In the first case a polynomial solution is obtained when the number of stochastic edges δ is small. The second case is when the number of realizations in the network is relatively small.

IV. Discussion

A. Gossiping and Learning

In this subsection we will illustrate the importance of gossiping by comparing the learning rates of the gossip and non-gossip travelers. We assume the G-DWC model with R possible realizations. When the traveler starts his journey he doesn't know what is the current network realization $r \in R$. Each time he gathers information about some edge weights he can eliminate zero or more network realizations which are inconsistent with the obtained weight. Depending on the network weights distribution, the traveler will be able to determine the current realization of the network after obtaining information about the state of enough edges. Since each time the traveler visits a vertex he gathers information about the state of all the emerging roads we define information vertices as the set of vertices the traveler needs to visit in order to find the current network realization, and denote it by k. In the following subsection we assume that the traveler doesn't visit a vertex more than once and that the information vertices are distributed uniformly at random in the network.

We first analyze the non-gossip traveler which we call a *Step-By-Step* (SBS) traveler, he receives information about a vertex only when he visits it. The probability that after i steps in the network (visiting i vertices) the SBS traveler already visited j out of the k information vertices is given by the hypergeometric distribution.

$$Pr(n,k,i;x=j) = \frac{\binom{k}{j}\binom{n-k}{i-j}}{\binom{n}{i}} \quad where \ j \le k; \ j \le i \le n$$

The probability that after visiting i vertices the SBS traveler already visited all k information vertices and thus found the current network realization is

$$Pr(n,k,i;x=k) = \frac{\binom{k}{k}\binom{n-k}{i-k}}{\binom{n}{i}} \quad where \quad k \le i \le n$$

The expected number of steps the SBS traveler needs to take to find all k information vertices is

$$\sum_{i=k}^{n} iPr(n,k,i;x=k) = \sum_{i=k}^{n} i\frac{\binom{n-k}{i-k}}{\binom{n}{i}}$$

Normalizing the above expression

$$\frac{\sum_{i=k}^{n} i\frac{\binom{n-k}{i-k}}{\binom{n}{i}}}{\sum_{i=k}^{n} \frac{\binom{n-k}{i-k}}{\binom{n}{i}}} = \frac{\sum_{i=k}^{n} i\frac{i!}{(i-k)!}}{\sum_{i=k}^{n} \frac{i!}{(i-k)!}} = \frac{(n+1)k+n}{2+k} \quad (17)$$

Eq. 17 indicates that the number of steps the SBS traveler needs to take in order to find the current network realization is proportional to the network size, n.

Unlike the SBS traveler that can only gather information about one new vertex in each step, the gossip traveler has additionally a probability to receive information about all the network's remaining unknown vertices. In his first step the gossip traveler receives information about $\xi_T n$ vertices and in the ith step about $\xi_T(1-\xi_T)^i n$ vertices. In each step the gossip traveler has information about all the vertices he learned about in his previous steps. Therefore, in the ith step the gossip traveler has information about $g(i)$ vertices:

$$g(i) = \sum_{j=0}^{i-1} \xi_T(1-\xi_T)^j n = (1-\bar{\xi_T}^i)n$$

where $\bar{\xi_T} = 1-\xi_T$.

Obviously, when the traveler gathers information about all n network's vertices he has information about all k information vertices and knows the network current realization. Thus, an upper bound on the expected number of steps the gossip traveler needs to take is the number of steps needed to gather information about all the network vertices. Since the number of vertices is discrete we are looking for the step number, r, such as

$$g(r+1) - g(r) = n(\bar{\xi_T}^r - \bar{\xi_T}^{r+1}) < 1$$

Solving the above equation yield

$$r < -\frac{\ln(n\xi_T)}{\ln(1-\xi_T)} \quad (18)$$

In practice in the gossip model r could be even smaller since the gossip traveler gather information by both gossiping and visiting vertices, however in the above analysis we took into account only gossiping. Thus, Eq. 18, is an upper bound on the expected number of steps the gossip traveler needs to take in order to find the current network realization. Comparing Eq. 18, to the expected number of steps the SBS traveler needs to take, Eq. 17, we conclude that the outcome of gossiping is higher learning rate. While the SBS traveler needs on average to visit O(n) vertices of the network to learn its state, the gossip traveler needs to visit only $O(\log(n))$ of them. In most cases higher learning rate in stochastic networks will result in shortest path to destination. Once the traveler knows the network edge's states he can reduce his path cost, for example by avoiding blocked roads.

B. *Characteristics of traveling in Gossip Networks*

In this section we will discuss the characteristics of optimal routing in gossip networks under the proposed GOSSIP_DP algorithm. For the simplicity of the discussion we use the following assumptions: The network is in the G-IWC model with one stochastic edge. The stochastic edge can be either in the “UP” or “DOWN” states. In the “UP” state the stochastic edge weight is similar to the weight of the deterministic edges, in the “DOWN” state its weight is higher than the weights of the deterministic edges. The traveler must traverse the stochastic edge on his way from source to destination. Once we analyze the parameters that influence routing under these assumptions expanding the model to the case of several stochastic edges with several stochastic states is straightforward as we demonstrate in the numerical analysis in the next section.

A traveler in the gossip networks that is navigating using our optimal routing policy can be viewed as operating in three different regimes: “WIN”, “LOSE”, and “NEUTRAL”. In the *“WIN”* regime the traveler reduces his travel cost by gossiping. In the *“NEUTRAL”* regime obtaining information doesn't increase or decrease the gossip traveler's path cost. In the *“LOSE”* regime obtaining information actually increases the traveler path cost. The operating regime is a result of the following parameters: the magnitude of the difference between the values of the actual weight of the stochastic edges ($\hat{w}_e$) and their expected weights (ω_{SE}), the values of the topology probability (ξ_T), and the magnitude of the difference between the values of the stochastic edges actual state (ξ_A) and a priori probability to be in the “UP” state (ξ_U) (see Table II for notation summary). Next we will explain the influence of each parameter.

The magnitude of the difference between the traveler's a priori knowledge (ω_{SE}) and the actual weight of the stochastic edges ($\hat{w}_e$), denoted by $\Delta_\omega = |\omega_{SE} - \hat{w}_e|$, determines the influence of obtaining information on the traveler's path cost. When ω_{SE} and $\hat{w}_e$ are similar, a gossip traveler will not have an advantage over a non-gossip traveler, they both know a priori the “correct” stochastic state. However, above some critical difference, $\Delta_\omega > \Delta_C$ obtaining information will decrease the

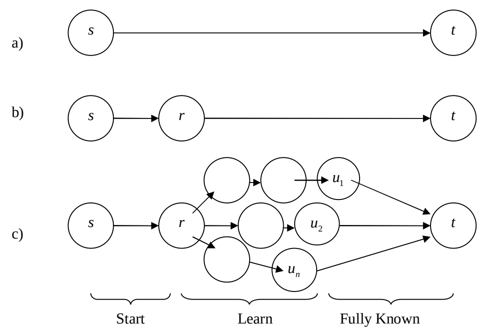

Fig. 5. The different possible paths a traveler can have for different topology probabilities. (a) No gossiping, (b) Maximal gossiping, and (c) In between.

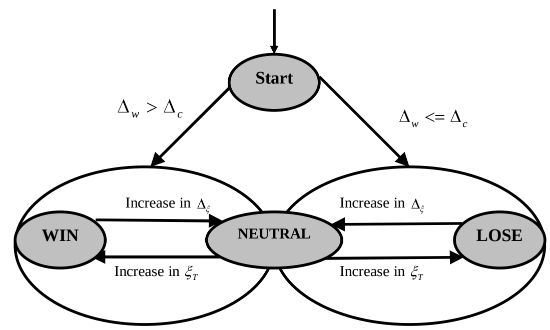

Fig. 6. The regime state diagram determines the influence of gossiping on routing in different network characteristics

traveler's path cost. For example, when ω_{SE} "tells" the travelers that a stochastic edges is in the "UP" state and the actual state is "DOWN" a non-gossip traveler may include this edge in its path while a gossip traveler will reduce his path cost by bypassing it in advance. The value of Δ_C is determined by the difference that will cause the non-gossip traveler to take the wrong path, meaning that he will bypass the stochastic edge when it's "UP" or travels through it when it's "DOWN".

Fig. 5 illustrates the different possible types of paths a traveler can have for different values of topology probability (ξ_T). When there is no gossiping (a) the probability to receive information is zero thus the optimal policy is determined a priori before the start of the journey and has no recourse. In this case the optimal policy is the one that minimize the expected weights. When ξ_T is maximal (b) the traveler learns about the state of all the stochastic edges on the traversal of the first edge, (s, r), and then travels to the destination t with full knowledge about $\hat{w}_e$ and therefore without changing his course. When ξ_T is in between (c) the traveler's path is composed of three phases, the initial phase is until the traveler obtains any information about the state of the stochastic edges. Then, in the learning phase, the traveler may recalculate and recourse his path according to the updated information vector – his optimal policy is a collection of different branches. When the traveler has full information about $\hat{w}_e$, at some vertex u, he travels to the destination without changing his course. The higher ξ_T the quicker the gossip traveler will learn about the state of the network and therefore minimize the learning phase in his travel which leads to decrease in the policy cost.

According to the optimal policy, stated in Eq. 9, one of the parameters that determines the relative weight of each branch in the path is the a priori probability of the network to be in certain stochastic state, denoted here by ξ_U. The closer ξ_U is to ξ_A (small $\Delta_\xi = |\xi_U - \xi_A|$) the more efficient the learning phase will be. Efficient learning means that the traveler is directed toward the "right" direction by giving higher relative weight to the "right" branch. When there is a relatively large difference between ξ_U and ξ_A the branches in the learning phase will direct the traveler to the "wrong" direction and as a result the cost of his policy will increase. For example, when the a priori probability of the stochastic edge to be in the "UP" state is small ($\xi_U \approx 0$) the optimal policy will direct the gossip traveler to branches that detour the stochastic edge. When the stochastic edge is actually in the "DOWN" state this decision is beneficial, however when the actual state of the stochastic edge is "UP" the decision will maximize the gossip traveler learning phase and his total traveling cost.

The operating regime that the traveler experiences is determined by the combined values of the parameters, Δ_ω, ξ_T, and Δ_ξ. Fig. 6 is a state diagram that illustrates the influence of the parameters on the network regime. When Δ_ω is below some threshold, Δ_C, the a priori knowledge of the network state is close enough to the true value, and thus increasing the path length to obtain information can not benefit the gossip traveler. As a results, in this case, the network can be either in the "NEUTRAL" or "LOSE" regimes. The "LOSE" regime is obtained when the learning phase is relatively large (increase in Δ_ξ), however a larger topology probability shortens the learning phase and pushes the network into the "NEUTRAL" regime. The ultimate network regime is determined by the relation between these two parameters ξ_T, and Δ_ξ. Similarly, when Δ_ω is above the threshold, Δ_C, gossiping helps the gossip traveler to reduce his policy costs. The network can be either in the "WIN" or "NEUTRAL" regimes according to the relation between ξ_T, and Δ_ξ. In the next section, we will demonstrate the above discussion using simulation results.

Notation	Description
ω_D :	Weights of the deterministic edges
ω_{SE}:	Expected weights of the stochastic edges
$\hat{w_e}$:	Actual weight of the stochastic edges
Δ_ω:	$\lvert\omega_{SE} - \hat{w_e}\rvert$
Δ_C:	Critical value of Δ_ω
ω_{SD} :	Weights of the stochastic edges in "DOWN" state
ξ_A :	Stochastic edges actual state
ξ_T :	Topology probability
ξ_U :	A priori probability of the stochastic edges to be in the "UP" state
Δ_ξ :	$\lvert\xi_U - \xi_A\rvert$
θ_E:	Expected cost of the optimal policy
θ_R:	Relative expected cost (θ_E) at some topology probability; $\theta_E(\xi_T)/\theta_E(0)$
θ_A:	The average of relative expected (θ_R) over the whole range of ξ_T
Configuration:	A set of values for the above parameters
Operation Regime:	Determined by the network configuration. Can be either "WIN", "NEUTRAL" or "LOSE"

TABLE II
NOTATION SUMMARY

V. NUMERICAL ANALYSIS

The main purpose of the simulations was to investigate the influence of gossiping on the traveler's optimal policy cost under the different parameters used in the gossip networks. The performance and behavior of the proposed algorithm on the gossip networks are examined through numerical experiments on various grid network configurations with random generated weights under the G-IWC model. In each network configuration the simulation derived results comparing the traveler's expected optimal policy cost for different topology probabilities.

First, for each randomly generated network configuration the optimal routing policy tables are calculated using the GOSSIP_DP algorithm. Then, using the calculated routing tables the simulation computes the expected optimal policy cost from each vertex to the destination. For notation of the parameters we use see Table II.

A. Simulation design

The simulation was conducted on fully connected grid networks representing, for example, the road structure in many urban areas. Fig. 7 shows such a network for a 4×4 grid. The weights of the different deterministic edges were selected uniformly at random. Three specific edges in the grid were chosen to be stochastic. The stochastic edges could be in two states, with probability ξ_U in the "UP" state, then the edge weights are randomly selected exactly like the deterministic edges. When the stochastic edges are in the "DOWN" state their weights are set to different values as explain further below. The stochastic edges were selected such that they will have a significant influence on the optimal policy to the destination vertex t. For the same reason, the weight of the deterministic edge that is adjacent to t was set to be higher than the other deterministic edges.

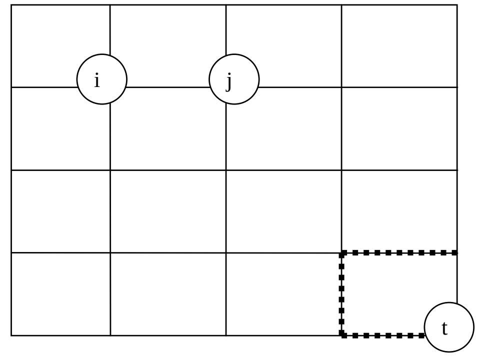

Fig. 7. A 4×4 grid network used in the simulations. The dashed lines are stochastic edges with probability ξ_U to be in the "UP" state. Larger grids had the same structure.

The following list details the range of values we used in the simulation:

Deterministic weight (ω_D) : Uniformly at random in [1,100].

Stochastic "DOWN" weight (ω_{SD}) : In each configuration all stochastic edges had the same weight which was selected uniformly at random in [0,800].

Topology probability (ξ_T) : In each configuration the same value of ξ_T was set to all the edges in the network. The range of tested values was in [0,1].

A priori probability (ξ_U) : Different values in the range [0,1] were used to test the influence of ξ_U. In each configuration all stochastic edges were set to the same value.

Stochastic actual state (ξ_A) : The actual state of all three stochastic edges was set equally to either "UP" or "DOWN".

Network structure (Grid Size) : Two different grid networks were used with sizes of 4×4 and 8×8.

Totally we tested $21(\xi_T) \cdot 9(\omega_{SD}) \cdot 11(\xi_U) \cdot 2(\xi_A) = 4158$ different configuration for each grid size.

In order to remove the influence of specific random network weights the same set of experiments were repeated with the same network configuration for ten different random seeds. The analyzed results are averaged over the ten different runs.

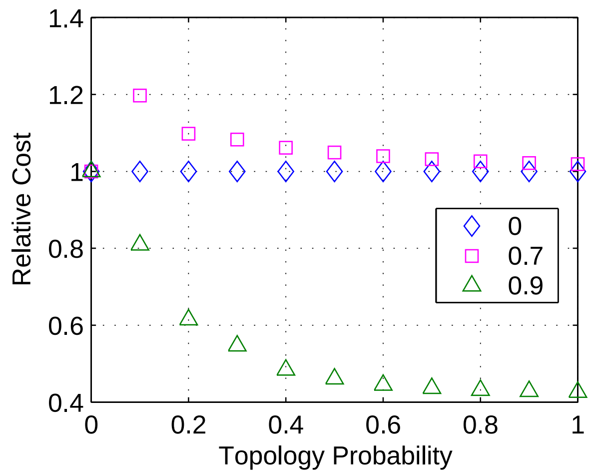

Fig. 8. The influence of topology probability (ξ_T) on path costs (θ_R) in different a priory probabilities (ξ_U). Simulation was done with grid size = 4 × 4, ξ_A = "DOWN", ω_{SD} = 700, ξ_U = 0, 0.7, and 0.9.

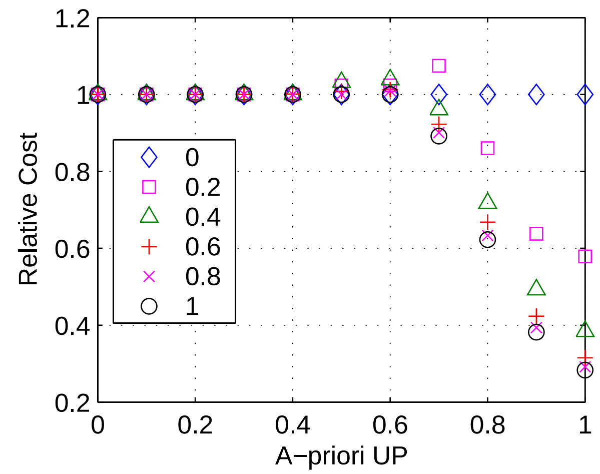

Fig. 9. The influence of a priory probability (ξ_U) on path costs (θ_R) in different gossip probabilities (ξ_T). The different graphs are drawn for $\xi_T = 0, 0.2, 0.4, 0.6, 0.8,$ and 1. Simulation was done with grid size = 4, ξ_A = "DOWN", and ω_{SD} = 600.

B. Performance Measurement

After the routing tables were built for a given network configuration the *Expected Cost* (θ_E) from each vertex to the destination was calculated. θ_E is calculated by following all the possible paths from source to destination assuming that the traveler starts his travel with no information $I = \{X, X, X\}$. The paths were weighted according to their probability to occur. The results are presented using the value of *Relative Expected Cost* (θ_R), where

$$\theta_R(\xi_T) = \frac{\theta_E(\xi_T)}{\theta_E(\xi_T = 0)}$$

When $\theta_R = 1$ gossiping doesn't change the gossip traveler's θ_E and we are in the "NEUTRAL" regime. For $\theta_R < 1$ obtaining information leads to a decrease in θ_E – the "WIN" regime. In the case of $\theta_R > 1$ obtaining information leads to an increase in θ_E, contradicting the desirable outcome – the "LOSE" regime. We are interested in the value of θ_R and less in the value of θ_E since we are mainly interested in the influence of obtaining information on the performance of a given network configuration.

Some of our results are presented using the values of θ_A which is the *Average* of θ_R over all the different measured gossip probabilities for a given network configuration.

C. Results Discussion

The results presented in Fig. 8 demonstrates the role of obtaining information in different network configurations. In this example ξ_A ="DOWN", thus when ξ_U=0 obtaining information doesn't change the traveler's optimal policy cost. When $\Delta_\omega = \Delta_\xi = 0$ obtaining information will not help the gossip traveler, both travelers are directed in the "right" direction and the gossip traveler has a minimal learning phase, as a result the network operates in the "NEUTRAL" regime. When ξ_U=0.7 obtaining information increases the traveler's optimal policy cost – the network is in the "LOSE" regime. In this case ω_{SE} is such that the non-gossip traveler bypass the stochastic edges, which is justified since ξ_A="DOWN". Therefore, the non-gossip traveler knows the "right" direction. Obtaining information only puzzles the gossip traveler due to Δ_ξ that implies that the learning phase will be relatively large, as a result the gossip traveler will increase his optimal policy cost. Increase in the ξ_T leads to shorter learning phase which leads to smaller θ_R. When ξ_U=0.9 the network is in the "WIN" regime. In this case $\Delta_\omega > \Delta_C$, thus the non-gossip traveler roam in the "wrong" direction. Increase in ξ_T leads to reduce in θ_R since the gossip traveler finishes his learning phase quicker. Fig. 8 also illustrates that the magnitude of the "WIN" effect is substantial larger than the "LOSE" effect.

Fig. 9 depicts the relation between ξ_U and θ_R for different ξ_T values. The curves move between three regimes. When ξ_U is below a threshold value, an increase in ξ_U doesn't change θ_R – the network is in the "NEUTRAL" regime. Then, an increase in ξ_U leads to an increase of θ_R and the network is in the "LOSE" regime. Further increase of ξ_U moves the network into the "WIN" regime. Comparing the graphs for different ξ_T reveals that in the "NEUTRAL" regime the behavior of all the graphs is almost identical. In the "LOSE"

Averaged Path Costs at Different Network Configurations for 4 x 4 Grid

A-priory "UP" \ "DOWN" Weight	0	100	200	300	400	500	600	700
1	0.973	0.973	0.851	0.728	0.640	0.569	0.516	0.476
0.9	0.973	0.973	0.897	0.790	0.728	0.678	0.709	0.653
0.8	0.973	0.972	0.914	0.847	0.814	0.739	0.774	0.868
0.7	0.973	0.976	0.927	0.883	0.816	0.888	0.970	1.071
0.6	0.973	0.994	0.975	0.863	0.964	0.958	1.049	1.044
0.5	0.988	0.985	0.972	0.935	0.966	1.030	1.029	1.029
0.4	0.988	0.993	0.966	0.989	1.012	1.012	1.012	1.012
0.3	1.000	0.993	0.991	1.002	1.002	1.002	1.002	1.002
0.2	1.000	1.000	0.998	1.001	1.001	1.001	1.001	1.001
0.1	1.000	1.000	1.000	1.001	1.001	1.001	1.001	1.001
0	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

"DOWN" Weight

Fig. 10. θ_A for different values of ω_{SD} (X axis) and ξ_U (Y axis). White cells represents the "WIN" regime, gray the "NEUTRAL" regime and darker gray the "LOSE" regime. This simulation was done with the following parameters: grid size = 4; ξ_A = "DOWN"; θ_A was averaged over ξ_T = 0 to 1.

regime, the θ_R peak is reached at ξ_T = 0.2. In the "WIN" regime increase in ξ_T leads to a decrease in θ_R.

In this graph the network is in the "NEUTRAL" regime when ω_{SE} and $\hat{w}_e$ are similar and the difference between ξ_A and ξ_U is small. In the "LOSE" regime the increase in Δ_ξ leads to a longer learning phase and as a result an increase in θ_R. In the "WIN" regime the increase in Δ_ξ increases the learning phase while an increase in ξ_T decreases it, however the non-gossip traveler moves towards the stochastic edge which increases his θ_E significantly compare to the θ_E of the gossip traveler. As a result, taking both parameters into account, the relative optimal policy cost of the gossip traveler, θ_R, is reduced.

Fig. 10 illustrates the relation between ξ_U and ω_{SD} for averaged ξ_T when the grid size is 4 X 4. Here are several observations from the results:

1) When ξ_U is zero, ω_{SE} is equal to ω_{SD}, in this case the traveler knows a priori $\hat{w}_e$ and there is no benefit in obtaining information – the network is in the "NEUTRAL" regime.

2) At lower ω_{SD} $(0 - 200)$ increase of ξ_U leads the network into the "WIN" regime. In this case the stochastic edges weights is similar to the weights of the deterministic edges, therefore information helps the gossip traveler to find the optimal path in the network and decrease his θ_A only moderately.

3) At higher ω_{SD} $(300-)$ an increase of ξ_U leads the network from the "NEUTRAL" to the "LOSE" and then to the "WIN" regime. In the "NEUTRAL" and "LOSE" regimes the non-gossip traveler bypass the stochastic edges, therefore in this case obtaining information doesn't help the gossip traveler. When $\xi_U > 0$, obtaining information actually increases the learning phase due to relatively large Δ_ξ and thus there is an increase in the θ_A. Then, with the increase in ξ_U the non-gossip traveler tries to travel through the stochastic edges which leads to increase in his path cost and decrease in θ_A of the gossip traveler that bypass the stochastic edge. The move from the "LOSE" to "WIN" regime is not due to the fact that the gossip traveler decreases his path cost, he actually increases it. However the non-gossip traveler increases his path cost even more due to the fact that now he doesn't bypass the stochastic edges.

4) At higher ω_{SD} $(300-)$, with the increase in ω_{SD} there is an increase in the size of the "LOSE" regime. The "LOSE" regime ends when the non-gossip traveler decides to travel through the stochastic edges. This is happening when his ω_{SE} reaches ≈ 200 which is the cost of bypassing the stochastic edges in this example.

5) At higher ω_{SD} $(300-)$, in the "LOSE" regime, the value of θ_A increases with the increase in ξ_U and doesn't change with the increase in ω_{SD}. This phenomenon is due to the parameter Δ_ξ, at higher ξ_U there is a higher probability to paths that lead to the "wrong" direction.

6) In the "WIN" regime, an increase in ω_{SD} leads to a decrease in θ_A. In higher ω_{SD} the non-gossip traveler travels through the stochastic edges that have increased weights, therefore the gossip traveler can reduce his path cost to a larger extent.

7) In the "WIN" regime, an increase in ξ_U leads to a decrease in θ_A. The change here is more moderate and is the result of two parameters. On the one hand, with the increase in ξ_U the difference between ω_{SE} and $\hat{w}_e$ is increased which leads to an increase in the non-gossip traveler path cost and a decrease in θ_A. On the other hand, an increase of ξ_U leads to an increase in the learning phase which leads to the opposite result of an increase in θ_A. The outcome of the two parameters is a total decrease in θ_A.

Fig. 10 illustrates that for this network configuration gossiping helps in more than half of the cases. In addition, the gain from gossiping is far greater, as much as 50% reduction of the expected path cost, compare to the possible loss which is only up to 7%. However, the fact that one can lose from trying to obtain information dictates the need to understand gossip networks behavior.

Fig. 11 illustrates that the "LOSE" regime is less significant in larger grid sizes. The reason is that in a small grid the number of steps to the destination is small therefore even one wrong step can lead to a significant increase in the path cost. In larger networks, where the number of steps is relatively large, the influence of wrong moves is smaller. In real life traffic applications the smaller grid size behavior is more likely due to the small number of options the traveler have especially when the network is in the "DOWN" state, ie. during congestion.

Averaged Path Costs at Different Network Configurations for 8 x 8 Grid

A-priory "UP" \ "DOWN" Weight	0	100	200	300	400	500	600	700
1	0.953	0.994	0.939	0.839	0.739	0.663	0.603	0.555
0.9	0.972	0.994	0.938	0.838	0.756	0.678	0.659	0.607
0.8	0.972	0.994	0.936	0.852	0.794	0.754	0.736	0.743
0.7	0.972	0.994	0.947	0.891	0.849	0.824	0.807	0.892
0.6	0.983	0.994	0.977	0.930	0.876	0.901	0.941	1.006
0.5	0.983	0.994	0.976	0.948	0.894	1.004	1.003	1.003
0.4	0.983	0.994	0.976	0.959	0.969	1.002	1.002	1.002
0.3	0.987	0.994	0.991	0.958	1.001	1.001	1.001	1.001
0.2	1.000	1.000	0.999	0.989	1.000	1.000	1.000	1.000
0.1	1.000	1.000	0.999	1.000	1.000	1.000	1.000	1.000
0	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Fig. 11. θ_A for different values of ω_{SD} (X axis) and ξ_U (Y axis). White cells represents the "WIN" regime, gray the "NEUTRAL" regime and darker gray the "LOSE" regime. This simulation was done with the following parameters: grid size = 8; ξ_A = "DOWN"; θ_A was averaged over ξ_T = 0 to 1.

VI. Conclusions and Future Work

This paper presents and studies a new model for information gathering in stochastic networks, the gossip networks. Gossiping could lead to some unusual phenomena, where the optimal routing policy may direct travelers to make a detour in order to gather information and minimize their expected path cost. The optimal traveling policy in gossip networks is expressed by a dynamic programming equation. Although the algorithm that solves the equations, GOSSIP_DP, is intractable in general, we present two special scenarios where the optimal solution is polynomial in respect to the network size. We analyze the relation between the parameters that influence gossiping and produce a state diagram that predicts the network regime. Gossip networks can operate in three regimes, in each regime gossiping has different effect on the traveler optimal path cost, "WIN" (reduce), "NEUTRAL" (doesn't change) and "LOSE" (increase). Numerical studies on gossip grid networks confirm the regime analysis. The numerical studies illustrate that in the grid networks we study, the "WIN" regime is larger that the "LOSE" regime, both in size and in magnitude and that the "LOSE" regime is more common in small networks.

A. Future Work

This research can be continued in several directions. First, one can study optimal ad-hoc communication exchange protocols, best fitted to vehicles traveling at medium or high speeds. A second direction is to examine optimal routing in gossip networks, e.g., it is interesting to look at the effect of gossiping in different network model, such as, time dependent network or models that take into account the interactions between agents and the macroscopic properties of the system. Another possible future direction involves developing general approximation algorithms that overcome the "curse of dimensionality" while using the gossip networks unique properties.

One of the dominant parameters of the GOSSIP_DP algorithm is the topology probability. Future work is needed to understand the influence of traffic and communication factors on its value, in particular, the influence of parameters such as node density, node velocity, and radio transmission range.

Acknowledgment

The authors wish to thank to the anonymous referees for their constructive comments.

References

[1] D. Bertsekas and R. Gallager, *Data networks*, 2nd ed. Prentice-Hall, 1992.

[2] "Traffic information by monitoring cellular networks," http://www.appliedgenerics.com.

[3] "Traffic information via FM radio," http://www.tmcforum.com.

[4] "Traffic information to telematic systems," http://www.trafficmaster.net.

[5] 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.

[6] R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D. S. J. De Couto, "CarNet: A scalable ad hoc wireless network system," in *the 9th ACM SIGOPS European workshop: Beyond the PC: New Challenges for the Operating System*, Kolding, Denmark, Sept. 2000.

[7] "Official charter of Mobile Ad-hoc Networks maintained by the IETF," http://www.ietf.org/html.charters/manet-charter.html.

[8] L. Briesemeister, "Group membership and communication in highly mobile ad hoc networks," Ph.D. dissertation, School of Electrical Engineering and Computer Science, Technical University of Berlin, Germany, Nov. 2001.

[9] S. Pallottino and M. G. Scutellà, "Shortest path algorithms in transportation models: classical and innovative aspects," in *Equilibrium and Advanced Transportation Modelling*, P. Marcotte and S. Nguyen, Eds. Kluwer, 1998, pp. 245–281.

[10] G. Polychronopoulos and J. Tsitsiklis, "Stochastic shortest path problems with recourse," *Networks*, vol. 27, pp. 133–143, 1996.

[11] G. Andreatta and L. Romeo, "Stochastic shortest paths with recourse," *Networks*, vol. 18, pp. 193–204, 1988.

[12] A. Orda, R. Rom, and M. Sidi, "Minimum delay routing in stochastic networks," *IEEE/ACM Transactions on Networking*, vol. 1, no. 2, pp. 187–198, 1993.

[13] S. Waller and A. Ziliaskopoulos, "On the online shortest path problem," *Networks*, vol. 40, no. 4, pp. 216–227, 2002.

[14] M. E. Ben-Akiva, H. N. Koutsopoulos, R. G. Mishalani, and Q. Yang, "Simulation laboratory for evaluating dynamic traffic management systems," *Journal of Transportation Engineering*, vol. 123, no. 4, pp. 283–289, 1997. [Online]. Available: http://link.aip.org/link/?QTE/123/283/1

[15] R. Parshani, "Routing in gossip networks," Master's thesis, Department of Mathematics and Computer Science, Bar Ilan University, Oct. 2004.

[16] R.-P. Schafer, K.-U. Thiessenhusen, E. Brockfeld, and P. Wagner, "a traffic information system by means of real-time floating-car data," in *ITS World Congress 2002*, Chicago, USA, October 11-14.

[17] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-Based Ad Hoc Routing," in *IEEE INFOCOM 2002*, New York, NY, June 23–27 2002.

[18] J. Kulik, W. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," *Wirel. Netw.*, vol. 8, no. 2/3, pp. 169–185, 2002.

[19] D. Braginsky and D. Estrin, "Rumor routing algorthim for sensor networks," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. ACM Press, 2002, pp. 22–31.

[20] M. Puterman, *Markov Decision Processes*. John Wiley & Sons, 1994.

[21] J. Birge and F. Louveaux, *Introduction to Stochastic Programming*. Springer-Verlag, 1997.

[22] D. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Athena Scientific, 2000.

[23] R. Bellman, *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.