Asymmetric Information Diffusion via Gossiping on Static And Dynamic Networks

Ercan Yildiz[1], Anna Scaglione[2], Asuman Ozdaglar[3]
Cornell University, UC Davis, MIT

Abstract— **In this paper we consider the problem of gossiping in a network to diffuse the average of a sub-set of nodes, called sources, and directing it to another sub-set of nodes in the network called destinations. This case generalizes the typical average consensus gossiping policy, where all nodes are both sources and destinations of the average of the nodes data. We first describe prior results we obtained on a static network topology and gossip policy, highlighting what conditions lead to the desired information flow. We show that, through semi-directed flows, this formulation allows to solve the problem with lower complexity than using plain gossiping policies. Inspired by these results, we then move on to design algorithms to solve the problem in the dynamic case. For the dynamic network scenario we derive conditions under which the network converges to the desired result in the limit. We also provide policies that trade-off accuracy with increased mixing speed for the dynamic asymmetric diffusion problem.**

I. Introduction

Transportation and diffusion of information in large networks is a fundamental problem in Network Science [1]–[4]. However, in many cases, distribution of aggregated information is more important than the diffusion of individual node information. In particular, one may gain significant insight about the structure of the network data by analyzing aggregate quantities such as mean, median, max, quantiles rather than processing whole data. For this reason, Kempe *et.al.* have studied near neighbor gossiping based protocols for diffusing aggregate information in networks [5]. These protocols generate local traffic only, calculate the desired quantity on the fly, and are known to be robust to node/link failures. Gossiping algorithms have been studied extensively in the context of *consensus*, where all of the nodes in the network will have access to the same information in the end [2], [6]–[8].

As usual, we model a static network with N nodes as a graph $G(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of *directed* edges. Given the edge (i, j), i is the tail and j is the head of the edge. We define the neighbor set of node i as $\mathcal{N}_i \triangleq \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$. A dynamic network, instead, is a random collection of $G(\mathcal{V}, \mathcal{E}_n)$, where the edges that can be activated $\mathcal{E}_n$ changes at random. In this paper we work on the solution for the following asymmetric information diffusion problem: Each node in the network has an initial scalar measurement (or some sort of information that can be represented as a scalar value) denoted by $x_i(0) \in \mathbb{R}$, $i \in \mathcal{V}$. There exists a non-empty subset of the vertices $\mathcal{D} \subseteq \mathcal{V}$ (*destination* nodes), which are interested in the average of the values of another subset of the vertices $\mathcal{S} \subseteq \mathcal{V}$ (*source* nodes). We assume that $\mathcal{D} \cap \mathcal{S} = \emptyset$, *i.e.*, a source node cannot be a destination node at the same time. We have studied this asymmetric information diffusion problem in [9], [10] for a static network and have shown that, unlike consensus problems, the feasible set of solutions forms a non-linear non-convex region, that is difficult to characterize (even determine whether the set if empty or not). In this work, we pave our way towards the construction of a solution for random dynamic networks. We first establish a connection between our problem on static networks and the problem of designing transition probabilities of Markov Chains with given absorption probabilities. This connection will help us introduce a topology based condition for the existence of a feasible solution. Moreover, under mild assumptions, we prove that our problem can be casted as multi-commodity flow problems with acyclicity constraints, amenable to an Integer Programming formulation. The class of semi-directed solutions that emerges through this formulation inspires our extension of the algorithm to dynamic networks.

Related studies are [11]–[14]. However, our problem differs from these models in the sense that source and destination sets are disjoint, and there may exist intermediate nodes which are neither sources nor destinations.

II. Problem Formulation

We first consider the static case, and seek solutions that are in the class synchronous gossiping protocol, with constant non-negative update weights:

$$x_i(t+1) = W_{ii} x_i(t) + \sum_{j \in \mathcal{N}_i} W_{ij} x_j(t), \ i \in \mathcal{V}, \qquad (1)$$

where t is the discrete time index, W_{ij} is the link weight corresponding to the edge (j, i). We note that if $j \notin N_i$, then W_{ij} is simply equal to zero since node i can not communicate with node j in one step. We will discuss how to extend our model into asynchronous models in Section VI.

If we define a matrix $W \in \mathbb{R}^{N \times N}$ such that $[W]_{ij} = W_{ij}$ and a vector $x(t) = [x_1(t), x_2(t), \ldots, x_N(t)]^T$, then (1) can be written in the matrix form as:

$$x(t+1) = W x(t) = W^{t+1} x(0). \qquad (2)$$

Given (2), the limiting behavior of the system is:

$$\lim_{t \to \infty} x(t) = \lim_{t \to \infty} W^t x(0) = W^{\infty} x(0),$$

where $W^{\infty} \triangleq \lim_{t \to \infty} W^t$, assuming it exists.

The approach which we pursue here is to design a W that produces the desired computation irrespective of what $x(0)$ is, and distributes the value to the destinations only. Since, we are only interested in calculating the average of the nodes values in $\mathcal{S}$ at all nodes in $\mathcal{D}$, we have the following inherent constraints on W^{∞}:

$$W^{\infty}_{jk} = \begin{cases} |\mathcal{S}|^{-1} & k \in \mathcal{S} \\ 0 & k \notin \mathcal{S} \end{cases}, \ \forall j \in \mathcal{D}, \tag{3}$$

where $|.|$ denotes the cardinality of its argument. In other words, the entries of the limiting weight matrix should be chosen such that destination nodes give source nodes' values $|\mathcal{S}|^{-1}$ weight (for averaging), and give non-source nodes' values 0 weight. Therefore, the rows of W^{∞} corresponding to destination nodes should converge to $|\mathcal{S}|^{-1}$ for the source node columns, and 0 otherwise. We note that the structure of the limiting matrix W^{∞} is significantly different than the one in the *network-wide consensus* case.

Given a network $\mathcal{G}$, source-destination sets $\mathcal{S}, \mathcal{D}$, we denote a matrix W which satisfies the sparsity constraints of $\mathcal{G}$ along with the structure in (3) as an *Average Value Transfer* (AVT) solution to the problem of computing along routes (CAR). Therefore, the goal of this study is to determine the feasibility of such a problem as well as construct an AVT solution for a given configuration.

In the rest of the paper, unless stated otherwise, we will omit the word *AVT* and denote an AVT solution as the solution. Therefore, our (in)feasibility definition considers only AVT solutions. Moreover, we limit ourselves to codes in the set of *nonnegative matrices* with finite powers, *i.e.*, $W \geq 0, W^{\infty} < \infty$ and $\geq, <$ represent elementwise inequality, in analogy with the average consensus gossiping (ACG) policies [6] because they ease the mapping onto dynamic policies.

In the following section, we will discuss necessary and sufficient conditions on feasible AVT solutions for CAR. These conditions were originally introduced in [9] and included here for the sake of completeness.

III. Feasible Codes

We first partition vertices $\mathcal{V}$ into three *disjoint* classes: M sensors that belong to the *source nodes* set, K sensors that belong to the *destination nodes* set, and L sensors that belong to neither source nodes nor destination nodes (called *intermediate* nodes). In other words, $|\mathcal{S}| = M$, $|\mathcal{D}| = K$ and $L = N - M - K$. Without loss of generality, we index the set of source nodes as $\{1, \ldots, M\}$, the set of destination nodes as $\{M+1, \ldots, M+K\}$, and the set intermediate nodes as $\{M+K+1, \ldots, N\}$. The following lemma is in order:

Theorem 1: Given a network $\mathcal{G}$, source and destination sets $\mathcal{S}$ and $\mathcal{D}$, a $W \geq 0$ matrix is an AVT solution to the CAR problem if and only if:

1) *W is in the form:*

$$W = \begin{bmatrix} A & 0 \\ B & D \end{bmatrix}, \tag{4}$$

$A \in \mathbb{R}^{M \times M}$, $D \in \mathbb{R}^{N-M \times N-M}$ *and* $M = |\mathcal{S}_S|$,

2) $\lim_{t\to\infty} W^t$ *exists and is finite,*
3) $\lim_{t\to\infty} \left[D^{t+1}\right]_{1:K} = 0$,
4) $\lim_{t\to\infty} \left[\sum_{l=0}^{t} D^{t-l} B A^l\right]_{1:K} = \frac{1}{M}\mathbf{1}\mathbf{1}^T$,

where $[A]_{1:K}$ are the first K rows of A, and $K = |\mathcal{S}_D|$.

The proof of the theorem can be found in [9]. We note that the partition A governs the communication among source nodes, B governs the flow from source nodes and to the rest of the network, D governs the communication among non-source nodes. The upper right corner of the matrix W is an all zeros matrix which, in return, implies that there can not be any information flow from non-source nodes to source nodes. Such a finding is not surprising since the destination set is only interested in the average of the source set, and the average will be biased if source nodes hear from non-source nodes. To be able to understand third and fourth constraints of the theorem given above, we first note that given the structure in (4), the limiting matrix will be in the form of:

$$\lim_{k\to\infty} W^k = \lim_{k\to\infty} \begin{bmatrix} A^k & 0 \\ \sum_{l=1}^{k} D^{t-l} B A^l & D^k \end{bmatrix}. \tag{5}$$

We note that we are simply forcing the inherent structure in (3) on the limiting matrix given in (5) to obtain third and fourth constraints of the theorem. Unfortunately, designing a feasible W is difficult, since the fourth constraint in Theorem 1 is non-convex with respect to the elements of the partitions D, B and A, and also consists of all non-negative powers of A, D. Moreover, it is not clear from Theorem 1 what kind of topologies do or do not have solutions for the CAR problem we posed. In the following section, by focusing on stochastic matrices, we will discuss the similarities between our problem and designing transition matrices for absorbing Markov chains as well as topology based necessary conditions for the feasibility of AVT solutions.

IV. Stochastic codes solving the AVT

In this section, we focus on AVT codes which satisfies $W\mathbf{1} = \mathbf{1}$ as well as Theorem 1. Since such condition combined with non-negativity, implies that W is a stochastic matrix, we will refer these codes as *stochastic* solutions. A stochastic matrix W which satisfies Theorem 1, is the transition matrix of a Markov Chain $\mathcal{M}$ whose state space equals $\mathcal{V}$ and whose probability of jumping from state i to state j in a single step is W_{ij}. We say that node i has *access* to node j, if for some integer t, $W^t_{ij} > 0$. Two nodes i and j, which have access to the other, are said to be *communicating*. Since communication is an equivalence relation, the set of nodes which communicate forms a *class*. A class which consists of only source nodes is called a *source class*.

We first note that, due to the structure of W in (4), each source class forms an absorbing class by itself (Since non-source nodes can not access source classes). For a given non-source node i and a source class $\mathcal{S}_{C^\star}$, the quantity $\sum_{j\in\mathcal{S}_{C^\star}} W^{\infty}_{ij}$ will be equal to the probability that the chain is absorbed by the source class $\mathcal{S}_{C^\star}$ given the fact that $\mathcal{M}$ has been initialized at node i [15]. Moreover, for a given source node $i \in \mathcal{S}_{C^\star}$ and $j \in \mathcal{D}$, the quantity $W^{\infty}_{ij} / \sum_{j\in\mathcal{S}_{C^\star}} W^{\infty}_{ij}$ is the frequency that the chain visits node i given the fact that $\mathcal{M}$ has been initialized at node j and it has been absorbed by the source class $\mathcal{S}_{C^\star}$ [15]. Given the discussion above, we conclude the following:

Remark 1: Constructing a stochastic AVT code is equivalent to designing a transition probability matrix W for a Markov chain $\{\mathcal{M}(t)\}_{t=0}^{\infty}$ on graph $\mathcal{G}$ with state space $\mathcal{V}$, where each source class forms an absorbing class. Moreover, for each destination node $j \in \mathcal{D}$ and each source class $\mathcal{S}_{C^\star}$, absorbtion probabilities should be chosen such that:

$$P(\mathcal{M}(\infty) \in \mathcal{S}_{C^\star} | \mathcal{M}(0) = j) = \frac{|\mathcal{S}_{C^\star}|}{M},$$

where $P(.)$ the probability of its argument. Moreover, for each source node $k \in \mathcal{S}_{C^\star}$,

$$P(\mathcal{M}(\infty) \in k | \left(\mathcal{M}(\infty) \in \mathcal{S}_{C^\star} | \mathcal{M}(0) = j\right)) = \frac{1}{|\mathcal{S}_{C^\star}|}.$$

Interestingly, the formulation given above does not simplify our design problem, since constructing transition probability matrices for complex chains with given stationary distributions is a notoriously open problem in the literature. But the equivalence relation is useful to bring a different perspective in the analysis of the AVT problem and the equivalence is key to prove the following lemma, proven in [16]:

Lemma 1: Consider a network $F(E)$ and the sets $\mathcal{S}_S$ and $\mathcal{S}_D$. Partition the network into two disjoint sets (P, P^c) such that there exists at least one source class-destination pair in both sides of the network. For a given partition, we denote the links going from one set to the other set as cut edges, i.e*, an edge (i, j) is a cut edge if $i \in P$ and $j \in P^c$, or $j \in P$ and $i \in P^c$. If there exists a feasible stochastic code, then there exits at least two (cut) edges between P and P^c for all such partitions.*

Lemma 1 implies that connectivity is not sufficient for the existence of a stochastic AVT code. This is interesting because connectivity is a sufficient condition for the existence of an ACG solution [2]. Thus, we conclude that demands on stochastic AVT codes are stricter than the ones for the consensus problems. We would like to note that, as stated in the hypothesis, the conditions in Lemma 1 is valid for the cases where there are at least two source classes in the network. For the scenarios where there is only one source class, Lemma 1 is not valid.

V. Partially directed AVT solutions

In the following, we will formulate an integer programming problem whose solution will be utilized to construct so called *partially directed* AVT solutions. These solutions belong to a subset of the AVT solutions given in Theorem 1. We first introduce the definition of a partially directed AVT solution:

Definition 1: Consider a network $\mathcal{G}$ and the sets $\mathcal{S}$ and $\mathcal{D}$. A code W is a partially directed AVT solution, if it satisfies Theorem 1 and each link on the network, except the links among the source nodes, can be utilized only in one direction, i.e.*, $W_{ij}W_{ji} = 0, \ \forall \ i, j \notin \mathcal{S}$.*

It should be clear from Definition 1 why these codes are called partially directed solutions, *i.e.*, communication is directed only among the non-source nodes. Before proposing our construction, we will introduce some necessary definitions. Without loss of generality, we assume that each source class has already converged to the average value of its members by using an average consensus protocol. We enumerate the source classes (arbitrarily ordered) and we define the set of source classes as $\mathcal{S}_C$. We also define $\mathcal{E}' \subset \mathcal{E}$ which contains all edges except the edges among the source nodes. For any given $U \subset \mathcal{V}$, we define a set of edges $\mathcal{E}(\mathcal{U}) = \{(i, j) \in \mathcal{E}' | i, j \in U\}$. In other words, $\mathcal{E}(\mathcal{U})$ is the set of edges whose end points belong to the set $\mathcal{U}$. A method to construct partially directed AVT codes is as follows [16]:

Lemma 2: Consider a network $\mathcal{V}$ and the sets $\mathcal{S}$ and $\mathcal{D}$. For all $k \in \mathcal{S}_C$ and $l \in \mathcal{D}$, we define a variable b_i^{kl} as:

$$b_i^{kl} = \begin{cases} 1, & \text{if } i = k, \\ -1, & \text{if } i = l, \\ 0, & \text{otherwise}. \end{cases}$$

Consider the following integer programming formulation:

$$\text{miminize} \quad \max_{k \in \mathcal{S}_C, l \in \mathcal{D}} \sum_{(i,j) \in \mathcal{E}'} z_{ij}^{kl}, \tag{6}$$

$$\text{subject to} \tag{7}$$

$$\sum_{j|(i,j) \in \mathcal{E}'} z_{ij}^{kl} - \sum_{j|(j,i) \in \mathcal{E}'} z_{ji}^{kl} = b_i^{kl}$$

$$i \in \{\mathcal{S}_C \cup \{M+1, \ldots, N\}\}, k \in \mathcal{S}_C, l \in \mathcal{D}, \tag{8}$$

$$z_{ij}^{kl} \le u_{ij}, \quad (i,j) \in \mathcal{E}', \ k \in \mathcal{S}_C, \ l \in \mathcal{D}, \tag{9}$$

$$u_{ij} + u_{ji} \le 1, \quad (i,j) \in \mathcal{E}', \tag{10}$$

$$\sum_{(i,j) \in E(U)} u_{ij} \le |\mathcal{U}| - 1, \quad \mathcal{U} \subset \mathcal{V}, \mathcal{U} \neq \emptyset, \tag{11}$$

$$z_{ij}^{kl} \in \{0, 1\}, \quad (i,j) \in \mathcal{E}', \ k \in \mathcal{S}_C, \ l \in \mathcal{D}. \tag{12}$$

If the integer program given above has a solution, namely, ${z^\star}_{ij}^{kl}$, $(i, j) \in E'$, $k \in \mathcal{S}_C$, $l \in \mathcal{D}$, we define y_{ij}^k as follows:

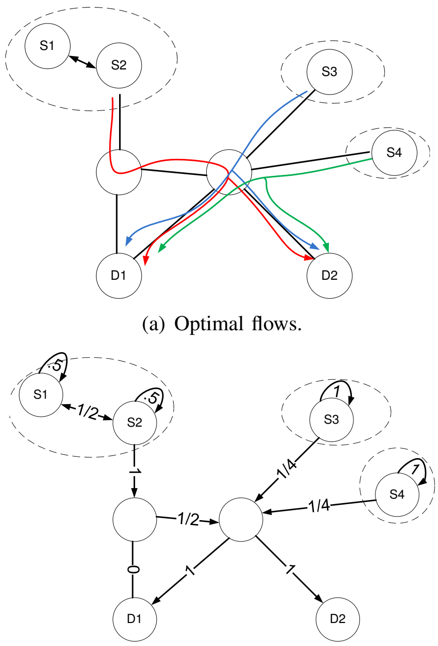
$$y_{ij}^k = \begin{cases} 1, & \text{if } \sum_{l \in \mathcal{S}} {z^\star}_{ij}^{kl} \ge 1, \\ 0, & \text{otherwise}. \end{cases}$$

Then, a feasible partially directed AVT code can be constructed as in (13)*, where $\mathcal{S}_{C_k}$ is the set of nodes which belongs to the k-th source class.*

The integer programming formulation given in the lemma is a directed multicommodity flow problem with acyclicity constraint [17]. In particular, one can map the variable b_i^{kl} to the net inflow at node i of data with origin k and destination l. The value of the net inflow is positive at the sources, negative at the destinations, and zero otherwise. The variable z_{ij}^{kl} indicates the amount of information with origin k and destination l that flows through link (i, j). y_{ij}^k is equal to one if there exists at least one flow on (i, j) that is originated from source class k.

We note that the constraint given in (8) guarantees the flow is preserved at each node, *i.e.*, the total inflow to a given node is equal to the total outflow from this node. Moreover, u_{ij} is a binary variable and equal to one if there exists at least one flow which utilizes the link (i, j) (9). Otherwise, it is equal to zero. Hence, (10) is the one-way flow constraint. Finally, (11) guarantees the flows are acyclic. The objective function of the problem is the maximum of number of paths between all source-destination pairs, thus the problem is minimizing the

$$W_{ji} = \begin{cases} \frac{\sum_{k\in\mathcal{S}_C} |S_C(k)| y_{ij}^k}{\sum_{l\in\mathcal{N}_j}\sum_{k\in\mathcal{S}_C} |S_C(k)| y_{lj}^k}, & \text{if } \sum_{l\in\mathcal{N}_j}\sum_{k\in\mathcal{S}_C} |S_C(k)| y_{lj}^k \neq 0 \text{ and } (i,j)\in\mathcal{E}', \\ \frac{1}{|\mathcal{N}_j|+1}, & \text{if } i,j\in\mathcal{S} \text{ and } i\in\mathcal{N}_j, \quad \text{or } j\in\mathcal{S} \text{ and } i=j, \\ 0, & \text{otherwise,} \end{cases} \tag{13}$$



(a) Optimal flows.

(b) Corresponding link weights.

Fig. 1. A directed solution. S and D represents source and destination nodes respectively.

convergence time of the directed part of the algorithm.

An example is given in Fig. 1. There are 4 source nodes with 3 source classes and 2 destination nodes. In Fig. 1(a), red, blue and green arrows represent flows from source classes to the destination nodes. The corresponding link weights are shown in Fig. 1(b). We note that the weight of the link connecting the flows from nodes 1 and 2 to the central hub is twice as much as the link weights of the other flows. This is because the number of source nodes in that particular flow is twice as much as the size of nodes in other flows.

We conclude that the existence of a partially directed AVT solution is a sufficient condition for the existence of an AVT solution, since partially directed AVT solutions also satisfy Theorem 1. On the other hand, the reverse argument may not always be true, *i.e.*, existence of an AVT solution does not imply the existence of a partially directed AVT solution. Remarkably, we were not able to find a counter example; we conjecture that the condition is both necessary and sufficient.

We also note that directed solutions are less robust to link/node failures while undirected solutions are more robust due to the presence of feedback. For instance, if a link on the path between a source-destination pair fails in a partially directed solution, the destination node will receive no information about that particular source . In the case of undirected solution, instead the destination nodes will still be able to obtain an approximate result.

VI. Extension to Dynamic Networks

In this section, we extend our model into the case where the underlying network is dynamic. One way to integrate the dynamical structure into gossiping algorithms is to consider an asynchronous policy, *i.e.*, nodes wake up at random times and perform random updates. Unlike *average* consensus algorithms, extension of the AVT codes into asynchronous policies is not straightforward. However, by considering the fact that non-source nodes simply mix and forward the values of the source nodes in the partially directed policies, we propose an asynchronous policy where source nodes keep averaging and non-source nodes keep swapping. Mathematically speaking, we assume that a single node $i \in \mathcal{V}$ is chosen at each discrete time instant $t \geq 0$ with probability $1/N$. The chosen node i selects one of its neighbors uniformly. Then, one of the following arguments hold:

1) If $i, j \in \mathcal{S}$, they simply average their values, *i.e.*, $x_i(t+1) = x_j(t+1) = 0.5x_i(t) + 0.5x_j(t)$.
2) If $i, j \notin \mathcal{S}$, they swap values, *i.e.*, $x_i(t+1) = x_j(t), x_j(t+1) = x_i(t)$.
3) If $i \in \mathcal{S}, j \notin \mathcal{S}$, they calculate a weighted average of their values, *i.e.*, $x_i(t+1) = x_j(t+1) = (1-\gamma)x_i(t) + \gamma x_j(t), \gamma \in [0, 1)$.
4) If $j \in \mathcal{S}, i \notin \mathcal{S}$, they calculate a weighted average of their values, *i.e.*, $x_i(t+1) = x_j(t+1) = (1-\gamma)x_j(t) + \gamma x_i(t), \gamma \in [0, 1)$.

We note that for $\gamma > 0$, the average of source nodes will be biased since a given source node i will hear from one of its non-source neighbors j. Since such a setup violates the first constraint in Theorem 1, one can not expect destination nodes to converge to the desired average. In the following, we will show that for $\gamma > 0$, the network will reach to a consensus in expectation.

Lemma 3: For a connected network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and the asynchronous policy defined above, for $\gamma \in (0, 1)$, the network will reach to a consensus in $\mathcal{L}^1$, i.e.

$$\lim_{t\to\infty} E\{x(t)\} = \alpha\mathbf{1}. \tag{14}$$

Proof: We first note that in the asynchronous case, the network will follow the update rule:

$$x(t+1) = W(t)x(t),$$

where the structure of $W(t)$ depends on the chosen node and its neighbor. For instance, if node i and j are the active ones at time $t \geq 0$, and if $i, j \in \mathcal{S}$, then:

$$W(t) = W_{kl}^{(ij)} = \begin{cases} 0.5 & \text{if } (k,l) = (i,j) \text{ or } (k,l) = (j,i) \\ 1 & \text{if } k = l \text{ and } k \neq i, j \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

We note that $W^{(ij)}$ denotes the matrix corresponding to the case where nodes i and j are active. One can construct $W^{(ij)}$ matrices for each of the four cases given above in a similar way. At this point, we define the average matrix W as:

$$W = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} W^{(ij)}. \tag{16}$$

W is the average of all possible $W^{(ij)}$ over all nodes in the network and their corresponding neighborhoods. $1/N$ is due to the fact that at each discrete time instant a node i is chosen with probability $1/N$, and node i chooses one of its neighbors uniformly randomly, *i.e.*, with probability $1/|\mathcal{N}_i|$. Since $W^{(ij)}\mathbf{1} = \mathbf{1}$ for all possible (i, j) pairs, W is a stochastic matrix. Moreover, since $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is connected, the average matrix W is irreducible. We also note that diagonals of W are all non-zero.

The expected value of $x(t), t \geq$ can be written as:

$$\lim_{t\to\infty} E\{x(t)\} = \lim_{t\to\infty} E\left\{\prod_{k=0}^{t} W(k)x(0)\right\} \tag{17}$$

$$= \lim_{t\to\infty} E\left\{\prod_{k=0}^{t} W(k)\right\} x(0) = \lim_{t\to\infty} W^t x(0), \tag{18}$$

where the last equality follows from the fact that node selection is i.i.d at each discrete index t and W is given in (16). Since W is stochastic, irreducible, and has non-zero diagonals, $\lim_{t\to\infty} W^t$ converges to an agreement matrix [18], *i.e.*, $\beta\mathbf{1}\mathbf{1}^T$ for some $\beta \in \mathbb{R}$. Thus, our result follows. ■

As we have mentioned, unfortunately, convergence to the true average is not guaranteed unless a very specific chain of edges are realized. One way to overcome this difficulty is to set $\gamma = 0$. In this case, source nodes will not hear from the rest of the network, thus the first constraint in Theorem 1 will not be violated.

Lemma 4: Given a connected network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and the asynchronous policy defined above, $\gamma = 0$, if for each source cluster-destination pair, there exists at least one path in between them which does not go through another source cluster, then, the node values will have a unique stationary distribution: Given there exists $|\mathcal{S}_C|$ source clusters in the network, and denoting their means as $\{\alpha_i\}_{i=1}^{|\mathcal{S}_C|}$ respectively, for a given node $j \notin \mathcal{S}$:

$$\lim_{t\to\infty} P(x_j(t) = \alpha_i) > 0 \quad \forall i \in \mathcal{S}_C, \tag{19}$$

$$\sum_{i \in \mathcal{S}_C} P(x_j(t) = \alpha_i) = 1. \tag{20}$$

Proof: Without loss of generality, we will assume that each source cluster consists of a single source node and each non-source node has been initialized with one of the source clusters' value. Due to the fact that nodes are chosen in an i.i.d. fashion at each iteration, the network states $\{x(t)\}_{t=0}^{\infty}$ forms a homogenous Markov Chain $\mathcal{M}$. In other words, given $x(t)$, $x(t+1)$ and $x(t-1)$ are independent. We denote the set of source clusters as $\mathcal{S}_C$. The state space of the chain $\mathcal{M}$ has $|\mathcal{S}_C|^{N-\mathcal{S}_C}$ elements, since each non-source nodes can assume $\mathcal{S}_C$ distinct values (due to swapping), and source cluster do not change their values at all (since $\gamma = 0$).

We denote the probability transition matrix of $\mathcal{M}$ as $P_{\mathcal{M}}$. Without loss of generality, let us assume that the chain has non-empty set of inessential states (it may be an empty set depending on the topology). Therefore, by potentially reordering the state space of $\mathcal{M}$, we can partition $P_{\mathcal{M}}$ as:

$$P_{\mathcal{M}} = \begin{bmatrix} P_1 & 0 \\ R & Q \end{bmatrix}, \tag{21}$$

such that $\lim_{k\to\infty} Q^k = 0$. We note that Q matrix represents the transition probabilities in between nonessential states. The set of indices corresponding to P_1 are the set of essential states. We will denote these indices as $\mathcal{M}_1$. In the following, we will prove that the set of essential states forms a single class, *i.e.*, if $y, z \in \mathcal{M}_1$, then $y \leftrightarrow z$. Note that both y and z correspond to an N-dimensional state vector in our model. Let's denote these state vectors as x_y and x_z. Let's define the set $\mathcal{D} = \{m : |[x_y]_m \neq [x_z]_m\}$, *i.e.*, the set of indices that are different in the configurations x_y and x_z.

For a given $m \in \mathcal{D}$, choose one of the source clusters with value $[x_y]_m$ or $[x_z]_m$ and determine a path from this particular cluster to m which does not go through any other source clusters. This has to be true, since otherwise, at least one of y and z will be inessential. Then, consider the following chain of events: The first non-source node on the path is chosen with $1/N$ probability and it selects the source node, and performs the update. Then, the second non-source node is chosen and it selects the first non-source node, so on. It should be clear that this particular event has non-zero probability for state vectors x_y and x_z. We repeat this particular argument *number of nodes on the path* times. At this point, all of the nodes on the path including the node m have values which are equal to the value of the chosen source cluster. Therefore, for a given $m \in \mathcal{D}$, this particular event will transform x_y and x_z into $x_{y'}$ and $x_{z'}$ respectively, where $\mathcal{D}' = \{m : |[x_{y'}]_m \neq [x_{z'}]_m\}$ is a *strict* subset of $\mathcal{D}$. If we apply the argument above for each $m \in \mathcal{D}$ sequentially, then x_y and x_z will be transformed into $x_{y^\star} = x_{z^\star}$.

We note that since y is an essential state and $y \to y^\star$, then $y^\star$ is also an essential state, *i.e.*, $y \leftrightarrow y^\star$. By the same way, one can show that $z \leftrightarrow z^\star$. Finally, since $y^\star = z^\star$ and essentiality is transitive, $y \leftrightarrow z$.

At this point, we have already shown that the chain $\mathcal{M}$ has a single essential class. Moreover, the essential class is aperiodic since there exists at least one aperiodic essential state, *i.e.*, each non-source equals to the same source node's value. The following lemma is due to Seneta [19]:

Lemma 5: Let $\mathcal{M}$ a Markov chain which has a single aperiodic essential class. Given its probability matrix $P_{\mathcal{M}}$ in the canonical form as in (21)*, define the stationary distribution to the primitive submatrix P_1 of $P_{\mathcal{M}}$ as v_1'. Let $v' = (v_1', 0')$ be an $1 \times N$ vector. Then as $k \to \infty$:*

$$P_{\mathcal{M}}^k \to \mathbf{1}v', \tag{22}$$

where v' is the unique stationary distribution corresponding to the chain $\mathcal{M}$ and $\mathbf{1}$ is the all ones vector.

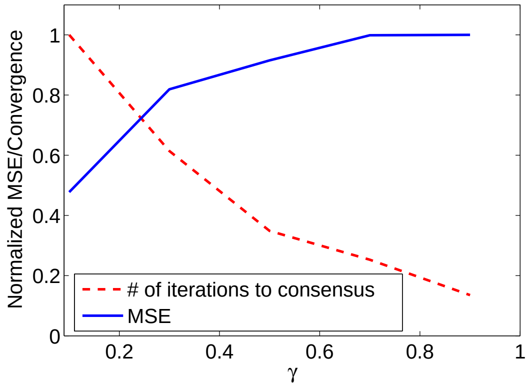
Since our $\mathcal{M}$ has a single aperiodic essential class, Seneta's Lemma implies that a stationary distribution exists and such a distribution is unique for our model.

Finally, we need to show that for each destination node j, there exists a set of essential states $\{y_i\}_{i=1}^{\mathcal{S}_C}$ where $[x_{y_i}]_j = \alpha_i$, for all $i \in \mathcal{S}_C$. In other words, the probability that a given destination node j being equal to a given source cluster i's value is non-zero. To see this, we only need to remind our readers that, by the hypothesis, for each destination node-source cluster pair, there exists at least one path which does not go through any other source clusters. Thus, we can find a chain of events that will change destination nodes values to a particular source cluster's value. This concludes our proof. ■

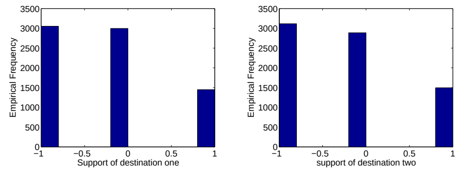
We note that if there is a single source cluster, the destination nodes will converge to that particular value. On the other hand, if there exists more than one source cluster, then, the network will have a unique stationary distribution which is independent of the initial values of non-source nodes. Moreover, each non-source node will keep switching its value forever, but, the support of the values it can switch is finite, and indeed these values are equal to the averages of each source cluster by (19)-(20). Therefore, in the long run, destination nodes start observing $|\mathcal{S}_C|$ distinct values, each of which are the averages of individual source clusters. Assuming all of the clusters have the same size, the true average can still be calculated by taking average of these observed quantities.

Finally, we would like to note the effect of the mixing parameter γ. As γ increases, the averages of distinct source clusters will mix faster due to increased information flow between the classes through the non-source nodes. But, the bias from the true average will also increase as γ increases. Thus, there is a clear trade-off between convergence speed and bias with respect to the γ parameter. At this point, we will leave the analysis of the convergence speed of the algorithm for both $\gamma > 0$ and $\gamma = 0$ as a future work, as well as the analytical characterization of the trade off and the complete characterization of the probabilities given in (19).

To justify our claims, we have simulated the proposed algorithm on random graphs. We have generated 100 geometric random graphs with $N = 50$ and connectivity radius $\sqrt{2 \log N/N}$. Three source nodes and three destination nodes are chosen randomly. For each graph, the algorithm is run for 5×10^5 steps. In Fig. 2(a), we plot the trade-off in between convergence speed/bias and the mixing parameter γ. As we have discussed above, the higher γ is, the faster network reaches an agreement. On the other hand, as γ increases, the bias from the initial mean of the source nodes increases. To validate Lemma 4, we have chosen $\gamma = 0$. Three source nodes are chosen such that all the assumptions in the lemma holds. The source nodes are given the values of $\{-1\}, \{0,\}, \{1\}$ respectively. Fig. 2(b)-2(c) shows the histogram of the two destination nodes in the long run. As



(a) MSE-Convergence speed versus γ.



(b) Support of destination one. (c) Support of destination two.

Fig. 2. Empirical Results.

we can see from these plots, destination nodes' values only fluctuates between source nodes' values.

VII. Conclusion

In this paper, we studied an asymmetric information diffusion problem where a set of destination nodes are interested in the average information of another set of source nodes. Utilizing gossiping protocols for the information exchange, we showed that our problem is equivalent to design transition probabilities for a Markov Chain on $\mathcal{V}$ with certain absorbing states. We argued that, under certain assumptions, our problem can be casted as multicommodity flow problem with acyclicity constraints. We proposed a simple asynchronous extension for dynamic networks.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, pp. 1204–1216, 2000.

[2] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, pp. 65–78, 2004.

[3] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, pp. 782–795, 2003.

[4] B. Krishnamachari, D. Estrin, and S. B. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proceedings of the 22nd International Conference on Distributed Computing Systems*, Washington, DC, USA, 2002, pp. 575–578.

[5] D. Kempe, A. Dobra, and J. Gehrke, "Computing aggregate information using gossip," in *Proc. Foundations of Computer Science*, Cambridge, MA, October 2003.

[6] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Info. Theory*, vol. 52, no. 6, pp. 2508–2530, June 2006.

[7] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Trans on Signal Processing, Accepted*, July 2009.

[8] A. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, “Geographic gossip: Efficient averaging for sensor networks,” *IEEE Transaction on Signal Processing*, vol. 56, no. 3, March 2008.

[9] M. E. Yildiz, A. Scaglione, and T. C. Aysal, “Computing along routes via gossiping,” in *Information Theory Workshop*, October 2009.

[10] M. E. Yildiz and A. Scaglione, “Directed gossiping for distributed data aggregation,” in *IEEE CAMSAP*, December 2009.

[11] D. Mosk-Aoyama and D. Shah, “Information dissemination via network coding,” in *Information Theory, 2006 IEEE International Symposium on*, July 2006, pp. 1748–1752.

[12] ——, “Fast distributed algorithms for computing separable functions,” *Information Theory, IEEE Transactions on*, vol. 54, no. 7, pp. 2997–3007, July 2008.

[13] U. A. Khan, S. Kar, and J. M. F. Moura, “Distributed sensor localization in random environments using minimal number of anchor nodes,” *IEEE Transactions on Signal Processing,Accepted*, 2008.

[14] F. Benezit, A. Dimakis, P. Thiran, and M. Vetterli, “Gossip along the way: order-optimal consensus through randomized path averaging,” in *Allerton*, 2007.

[15] D. Aldous and J. Fill, “Reversible markov chains and random walks on graphs,” http://www.stat.berkeley.edu/ aldous/RWG/book.html.

[16] M. E. Yildiz and A. Scaglione, “Computing along routes via gossiping,” *Accepted. IEEE Tran. on Signal Processing*, January 2010.

[17] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*. Athena Scientific, 1997.

[18] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, Cambridge, 1987.

[19] E. Seneta, *Non-negative Matrices and Markov Chains*. Springer Series in Statistics, 1981.