Network gossip algorithms

Devavrat Shah
Laboratory for Information & Decision Systems
Massachusetts Institute of Technology
Email: devavrat@mit.edu

***Abstract*— Unlike the Telephone network or the Internet, many of the next generation networks are not *engineered* for the purpose of providing efficient communication between various networked entities. Examples abound: sensor networks, peer-to-peer networks, mobile networks of vehicles and social networks. Indeed, these emerging networks do require algorithms for communication, computation, or merely spreading information. For example, estimation algorithms in sensor networks, broadcasting news through a peer-to-peer network, or viral advertising in a social network. These networks lack infrastructure; they exhibit unpredictable dynamics and they face stringent resource constraints. Therefore, algorithms operating within them need to be extremely simple, distributed, robust against network dynamics, and efficient in resource utilization.**

Gossip algorithms, as the name suggests, are built upon a *gossip* or *rumor* style unreliable, asynchronous information exchange protocol. Due to their immense simplicity and wide applicability, this class of algorithms has emerged as a canonical architectural solution for the next generation networks. This has led to exciting recent progress to understand the applicability as well as limitations of the Gossip algorithms. In this survey, I will discuss some of these recent results on Gossip network algorithms. The algorithmic results described here in a natural way bring together tools and techniques from Markov chain theory, Optimization, Percolation, Random graphs, Spectral graph theory, and Coding.

I. Introduction

The twentieth century has seen a revolution in terms of our ability to communicate at very long distances at very high speeds. This has fundamentally changed they way we live in the present world. The development of reliable and high-performance massive communication networks has been at the heart of this revolution. The telephone networks and the Internet are prime examples of such large networks. These networks were carefully engineered (and are still being engineered) for the single purpose of providing efficient communication given the available resources. In contrast to these networks, there has been a sudden emergence of different types of large networks in the past few years where the primary purpose is not that of providing communication. Examples of such networks include sensor networks, peer-to-peer (P2P) networks, mobile ad-hoc networks, and social networks.

A sensor network, made of a large number of unreliable cheap sensors, is usually deployed for the purpose of 'sensing', 'detecting' or 'monitoring' certain events. For example, smoke sensors capable of wireless transmission deployed for smoke detection in a large building, or a collection of interconnected camera sensors deployed for surveillance in a secure facility. The ability to deploy such networks anywhere with minimal cost of infrastucture has made them particularly attractive for these applications. Clearly, the primary purpose of such networks is to collect and process the sensed information by sensors rather than provide efficient communication.

The peer-to-peer networks are formed by connecting various users (*e.g.*, computers or handheld devices) over an already existing network such as the Internet. Usually such networks are formed with minimal infrastructural support. The peers (or neighbors) are connected over an existing network and hence the advantage of using such networks is not in terms of efficiency of utilizing resources. However, a significant benefit arises in terms of reduced infrastructural support in situations like wide information dissemination. For example, in the absence of a P2P network an Internet content provider (*e.g.*, BBC) needs to maintain a high bandwidth 'server farm' that 'streams' a popular movie or a TV show to a large number of users simultaneously. In contrast, in the presence of a P2P network a user is likely obtain the desired popular content from a 'nearby' peer and thus distributing a large cost of ' streaming' from the 'server farm' to many 'peers'. Therefore, such an architecture can reduce the cost of content dissemination for a content provider drastically. Of course, it is likely to come at an increased cost of the network utilization. Now, whether or not the benefits obtained in terms of reduced infrastructure by utilizing P2P network for a content

978-1-4244-2354-5/09/$25.00 ©2009 IEEE

provider offset the increased network cost incurred by the network provider is indeed intriguing both in an engineering and an economic sense. While the recent trend suggests that it is indeed the case (*e.g.*, advent of the BBCiPlayer [2] and adaptation of Korean ISPs [1]), the equilibrium solution is yet to be reached. Similar algorithmic issues arise in the context of mobile ad-hoc network formed between vehicles or future smart cars for the purpose of co-ordination, consensus or flocking (*e.g.*, see classical work by Tsitsiklis and co-authors [14], more recently [4], [7]).

Finally, we have noticed a very recent emergence of massive social networks between individuals connected over a heterogenous collection of networks. Until recently, an individual's social network usually involved only a small number of other acquintances, relatives or close friends. However, the arrival of 'social network applications' (*e.g.*, Orkut, Facebook, etc.) have totally changed the structure of existing social networks. Specifically, the social network of an individual now includes many more acquintances than before thanks to these online applications. Furthermore, the use of handheld devices like smart phones are likely to create new ways to 'socialize' through P2P networks formed between them in the near future. Naturally, this 'globalization' and 'ubiquitous presence' of social networks brings many exciting opportunities along with extreme challenges. To realize these opportunities and to deal with the challenges, we will need new algorithms with efficient effective social communication under uncertain environmental conditions.

A. NextGen networks: through an algorithmic lens

Algorithms are key building blocks of any network architecture. For example, the Internet provides efficient communication between users through a collection of algorithms operating at the end-users and inside the network. Popular instances of such algorithms are the Transport Control Protocol (TCP) for congestion control or Border Gateway Protocol (BGP) for routing. The above discussed emerging or next generation networks are not designed to provide efficient communication between the entities or the users networked by them. But, they do require algorithms to enable their primary applications. For example, a sensor network may require an estimation algorithm for event detection given the sensor observations; a P2P network may require a dissemination algorithm using peer information; a network of aerial vehicles may need an algorithm to reach consensus to co-ordinate their surveillance efforts, and an advertiser may need a social network algorithm for efficient 'viral' advertisement.

In most of these next generatio networks, algorithms usually need to operate under an 'adverse' environment. First of all, since these networks are not build for providing communication, there is usually a lack of a reliable network infrastructure. Second, these networks are highly dynamic in the sense that nodes may join the network, leave the network, or even become intermittently unavailable in an unpredictable manner. Third, the network is usually highly resource constrained in terms of communication, computation and sometimes energy resources.

The highly constrained environment in which algorithms are operating suggest that the algorithm must posses certain properties so as to be implementable in such networks. Specifically, an algorithm operating at a node of the network should utilize information 'local' to the node and should not expect any static infrastructure. It should attempt to achieve its task iteratively and by means of asynchronous message exchanges. The algorithm should be robust against the network dynamics and should not prescribe to any 'hard-wired' implementation. And finally, the algorithm should utilize minimal computational and communication resources by performing few logical operations per iteration as well as require light-weight data structures. These constraints naturally lead to 'Gossip' algorithms, formally described next, as a canonical algorithmic architectural solution for these next generation networks.

B. The formal agenda

Let us consider a network of n nodes denoted by $V = \{1, \dots, n\}$. Let $E \subset V \times V$ denote the set of (bidirectional) links along which node pairs can communicate. Let this network graph be denoted by $G = (V, E)$. This network graph G should be thought of as changing over time in terms of V and E. We model dynamics (or uncertainty) in the network by means of a stochastic probability matrix $P = [P_{ij}]$ where P_{ij} indicate probability that a node i can communicate to node j in a given time slot. Specifically, we will impose constraint that in a given time slot a node can communicate with at most one other neighbor. The performance of algorithm will be characterized in terms of the graph topology G and the dynamics matrix P. Finally, let d_i denote the degree of node i in G, *i.e.*, $d_i = |\mathcal{N}(i)|$, where $\mathcal{N}(i) = \{j \in V : (i, j) \in E\}$. Without loss of generality, assume G to be connected (under P).

We consider a class of algorithms, called 'Gossip' algorithms, that are operating at each of the n nodes of the network. Now we present the formal definition of these algorithms.

Definition 1 (Gossip algorithms): Under a Gossip algorithm, the operation at any node $i \in V$, must satisfy the following properties. (1) The algorithm should only utilize information obtained from its neighbors $\mathcal{N}(i) \stackrel{\triangle}{=} \{j \in V : (i,j) \in E\}$. (2) The algorithm performs at most $O(d_i \text{poly}(\log n))$ amount of computation per unit time. (3) Let $|F_i|$ be the amount of storage required at node i to generate its output. Then the algorithm maintains $O(\text{poly}(\log n) + |F_i|)$ amount of storage at node i during its running. (4) The algorithm does not require synchronization between node i and its neighbors, $\mathcal{N}(i)$. (5) The eventual outcome of the algorithm is not affected by 'reasonable' that allow for a possibility of eventual computation of the desired function in a distributed manner. changes in $\mathcal{N}(i)$ during the course of running of the algorithm.

We wish to design Gossip algorithms for computing a generic network function. Specifically, let each node have some information, and let x_i denote the information of node $i \in V$. The node $i \in V$ wishes to compute a function $f_i(x_1, \ldots, x_n)$ using a Gossip algorithm. Also, it would like to obtain a good estimate of $f_i(x_1, \ldots, x_n)$ as quickly as possible. The question that is central to this survey is that of indentifying the dependence of the computation time of the Gossip algorithm over the graph structure G and the functions of interest $f_1, \ldots, f_n$. Now some remarks.

First, property (3) rules out 'trivial' algorithms like *first collect values $x_1, \ldots, x_n$ at each node and then compute $f_i(x_1, \ldots, x_n)$ locally* for functions like summation, *i.e.*, $f_i(x_1, \ldots, x_n) = \sum_{k=1}^n x_k$. This is because for such a function the length of the output is $O(1)$ (we treat storage of each distinct number by unit space) and hence collection of all n items at node i would require storage $\Omega(n)$ which is a violation of property (3). Second, the computation of complex function (e.g. requiring beyond poly$(\log n)$ space) are beyond this class of algorithms. This is to reflect that the interest here is in functions that are easily computable, which is usually the case in the context of network applications. Third, the definition of a Gossip algorithm here should be interpreted as a rough guideline on the class of simple algorithms that are revelant rather than a very precise definition.

II. Summary of results

Given the setup described above, we provide a brief summary of some of the known results. We will describe a collection of network problems for which good Gossip algorithmic solutions are known. We will provide qualitative description of these solutions and refer reader to relevant papers for precise description of algorithm and exact statements of the results. The description of the solutions is in terms of "network layers". First, we describe the most basic networking task of designing reliable transport layer over the unreliable network (or information dissemination) under gossip constraints. Next, we describe a gossip algorithmic result for computation of linear functions (or averaging) that operate over the unreliable transport. This is followed by the task of computation of separable functions. Gossip lgorithm for this builds on the information dissemination mechanism. Finally, we describe results for two very important network tasks that are obtained through gossip algorithms that utilize the separable function computation as a "subroutine": (a) network scheduling and (b) network convex optimization. Thus, there is a natural 'layering' of gossip algorithms available for solving an important and large class of network problems.

Information dissemination. The next generation networks, modeled through a probabilistic graph G (along with probability matrix P), have unreliable transport. Two basic scenarios for information spreading in such networks are: (a) one-to-many, i.e. one node broadcasts its information to all nodes; and (b) many-to-many, i.e. all nodes wish to broadcast their information to all nodes.

For the case of one-to-many, the natural gossip algorithm for spreading one piece essentially takes $O(\log n / \Phi(P))$ time to spread information to all nodes with high probability (see [12], [13]). Here, $\Phi(P)$ is the conductance of P defined as

$$\Phi(P) \quad = \quad \min_{S \subset V : |S| \leq n/2} \frac{\sum_{i \in S, j \in V \setminus S} P_{ij}}{|S|}. \qquad (1)$$

For the case of many-to-many, the gossip algorithm that uses natural gossip algorithm with a natural diligent schedule at each node or uses coding for transmission can spread information to all nodes in time essentially $O(\log n / \widehat{\Phi}(P))$ with high probability (see [12]). Here, $\widehat{\Phi}(P)$ is a 'conductance-like' property of graph G, P defined as

$$\hat{\Phi}(P) = \sum_{k=1}^{n-1} \frac{k}{\Phi_k(P)},$$

where the k-conductance $\Phi_k(P)$ is defined as

$$\Phi_k(P) \quad = \quad \min_{S \subset V : |S| \leq k} \frac{\sum_{i \in S, j \in S^c} P_{ij}}{|S|}.$$

Linear computation. The basic distributed estimation question boils down to computation of certain (weighted) average $\mathbf{x}_{\text{ave}} = (\sum_{i=1}^{n} x_i)$. A randomized algorithm based on matrix P was introduced in [5]. The computation time of this algorithm scales as $\Theta(\mathrm{T}_{\text{mix}}(\overline{P}) + \log n)$ with high probabiity where $\overline{P} = (P + P^T)/2$ and $\mathrm{T}_{\text{mix}}(\overline{P})$ is the mixing time of the Markov chain with transition matrix $\overline{P}$ (see [5] for detailed definitions). Now if graph G, under $\overline{P}$ has good 'expansion' properties (i.e. $O(\log n)$ mixing time) such as complete graph or expander graph, then the computation time is essentially minimal and essentially of the order of the diameter of the graph. However, if the graph has 'geometry' (e.g. ring graph or two dimensional grid graph), then the mixing time of symmetric matrix $\overline{P}$ can be much larger than its diamater (for ring graph, its n^2 or square of its diameter). For such graphs, using non-reversible random walk based construction, [9] devised algorithm that has computation time $O(\text{diameter})$ for any graph G but with an added cost of expansion of the graph topology. Also see [6] for use of geography to accelerate such algorithm.

Separable function computation. Separable function computation, or equivalently computation of summation of n numbers is a key network computational problem (two of its applications are described soon after). Certain *extremal* property of Exponential distribution allows *converting* the summation problem into minimum computation problem. This allows for an algorithm based on one-to-many information disseminaton to compute ε approximation of the summation in time $O(\varepsilon^{-2} \log n/\Phi(P))$ (see [13] for details) with high probability[1]. Indeed, there is an interesting 'quantization' of this algorithm that incurs additional $O(\log n)$ factor in time. Further, this algorithm is optimal in terms of its dependence on the graph structure G, P (see [3] for details on quantization and optimality).

Two applications. The summation or separable function computation algorithm leads to Gossip algorithm design for seemigly two complex network applications: (1) Scheduling in a constrained queueing network, e.g. scheduling in a multi-access wireless network or scheduling in a switch (see [10] and [8] for details). (2) Network convex optimization, e.g. resource allocation or routing in the Internet under flow-level model (see [11] for details).

III. Future directions

In this survey, I have attempted to advocate the Gossip or more generally message-passing as an algorithmic architecture for the next generation networks. And this is only the beginning. The scope of such algorithms is wider than discussed here. For example, the tremendous success of belief propagation style inference or estimation algorithms for coding, image processing or more recently in designing algorithms network hardware. A systematic understanding of Gossip algorithm in terms their design, applicability and limitations, in this broadly defined sense, is one of the most important future direction of research.

Acknowledgment. Author thanks NSF CNS, CCF and HSD divisions for support for this work and Ola Ayaso, Stephen Boyd, Munzer Dahleh, Arpita Ghosh, Kyomin Jung, Damon Mosk-Aoyama, Balaji Prabhakar, Tim Roughgarden and Jinwoo Shin for collaborations that has led to this survey.

References

[1] http://www.reuters.com/article/pressrelease/idus55597+16-jan-2008+bw20080116.

[2] www.bbc.co.uk/iplayer.

[3] O. Ayaso. *Information Theoretic Approaches to Distributed Function computation*. PhD thesis, Massachusetts Institute of Technology, 2008.

[4] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Joint 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05)*, December 2005.

[5] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE/ACM Trans. Netw.*, 14(SI):2508–2530, 2006.

[6] A. G. Dimakis, A. Sarwate, and M. Wainwright. Geographic gossip : Efficient aggregation for sensor networks. In *5th International ACM/IEEE Symposium on Information Processing in Sensor Networks (IPSN '06)*, April 2006.

[7] A. Jadbabaie, J. Lin, and A. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Autom. Control*, 48(6):988–1001, 2003.

[8] K. Jung and D. Shah. Low delay scheduling in wireless network. In *IEEE ISIT*, 2007.

[9] K. Jung, D. Shah, and J. Shin. Minimizing rate of convergence for iterative algorithms. *Submitted to IEEE Transaction on Information Theory*, 2008.

[10] E. Modiano, D. Shah, and G. Zussman. Maximizing throughput in wireless network via gossiping. In *ACM SIGMETRICS/Performance*, 2006.

[11] D. Mosk-Aoyama, T. Roughgarden, and D. Shah. Fully distributed algorithms for convex optimization problems. In *International symposium on distributed computation (DISC)*, 2007.

[12] D. Mosk-Aoyama and D. Shah. Information dissemination via network coding. In *IEEE ISIT*, 2006.

[13] D. Mosk-Aoyama and D. Shah. Fast distributed algorithms for computing separable functions. *IEEE Transaction on Information Theory*, 54(7):2997–3007, 2008.

[14] J. Tsitsiklis. Problems in decentralized decision making and computation. *Ph.D. dissertation, Lab. Information and Decision Systems, MIT, Cambridge, MA*, 1984.

[1] With high probability means probability at least $1 - 1/\text{poly}(n)$.