Peer-to-Peer Compressive Sensing for Network Monitoring

Ali Fattaholmanan, Hamid R. Rabiee, *Senior Member, IEEE*, Payam Siyari, Ali Soltani-Farani, and Ali Khodadadi

***Abstract*—Monitoring large-scale networks is a critical yet challenging task. Enormous number of nodes and links, limited power, and lack of direct access to the entire network are the most important difficulties. In applications such as network routing, where all nodes need to monitor the status of the entire network, the situation is even worse. In this letter, a collaborative model in which nodes pick up information from measurements generated by other nodes is proposed. Using this model, for the first time, an upper bound is derived for the number of measurements that each node must generate, such that the expected number of measurements observed by each node is sufficient to provide a global view of the entire networked data. Finally, by using this upper bound, an efficient optimization method is introduced to minimize the total number of measurements. The feasibility and accuracy of the proposed method is verified through extensive numerical simulations.**

***Index Terms*—Network monitoring, compressive sensing, peer-to-peer networks, network tomography.**

I. Introduction

PERIODICALLY monitoring the state of a network, such as links' loss rate in an ISP or environmental temperature in a Wireless Sensor Network (WSN), is an essential and challenging task for network administrators. Many works have been devoted to finding techniques that can accurately monitor large-scale networks with low communication overhead. The numerous number of entities to be monitored, lack of direct access to the entire network, and limited communication power of nodes in WSNs, are some of the challenging issues that highlight the need for an efficient method for monitoring networks. Undoubtedly, the recent application of Compressive Sensing (CS) theory in network inference problem [1]–[4], has opened a new chapter in the literature of network monitoring algorithms. By using the CS theory, it is possible to infer and monitor the entire network with only a few aggregate end-to-end measurements [2], [3].

Among various network monitoring algorithms, *distributed* network monitoring has a wide range of applications. In this category, all network entities need to periodically monitor the status of the entire network. As an example, for an Autonomous System (AS), identification of heavy loaded or congested links and changing the routing tables accordingly, results in balancing the traffic load among AS internal links which has a key role in providing a quality of service aware service for the users. This cannot happen unless the internal routers in the AS, occasionally monitor the load and performance of the links. However, using a central monitoring entity, not only forms a single point of failure in the system, but also leads to congestion of nearby links and routers.

In this letter, we address the problem of providing an accurate global view of the entire network for all the nodes/hosts in a large-scale network, by using the CS theory. We propose an efficient distributed method in which all the nodes cooperate with each other in order to monitor the entire network. In this context, the measurements are done by using a few end-to-end independent additive probes. Previously, Xu *et al.* [2] proposed an efficient *central* method for *network tomography*, defined as "link-level parameter estimation based on end-to-end, path-level traffic measurement" [5], using only a few number of measurements proportional to the network size. Xu *et al.* theoretically proved that $O(k \log n)$ measurements suffices to recover any k-sparse *networked data*, in a network of n nodes. In this method, a central server (e.g., sink in WSNs) is responsible for starting the measurements. The server will then use those measurements and their observed accumulated values to locally recover the unknown networked data.

Here, inspired by [6], we propose a new *distributed* framework in which nodes cooperate with each other to reduce the number of required measurements. In the proposed method, compared with the settings of [2], each node can be considered both as a server that starts the measurements, and as a client that responds to measurement requests. Consequently, at the end of the measurement, all nodes have access to the entire data, while in [2], it is assumed that only one central node has access to the networked data. Continuing our previous AS example, using our framework, all the routers within the AS can determine the congested links independently. Based on this information, the routers can change their routing table with the aim of alleviating the traffic load of the internal links of the AS. The distributed nature of this framework is somehow similar to the peer to peer (P2P) paradigm and thus, we use *peer* and *node* for the same concept, throughout this letter.

In addition, we theoretically derive an upper bound for the number of measurements that each node must generate. We show that if this upper bound is met, the expected number of measurements observed by each node will be sufficient to provide a global view of the entire networked data. Furthermore, we use this information to minimize the total number of measurements generated in the network, and determine how many measurements each peer must generate.

Finally, we have run extensive simulations to validate the performance of our analytical model against the previous state-of-the-art techniques. The obtained results show that compared with the state-of-the-art, our P2P framework requires a significantly smaller number of measurements to achieve a negligible recovery error. Moreover, the experiments reveal interesting results that confirm our intuition and analytical findings.

Manuscript received June 15, 2014; accepted September 5, 2014. Date of publication September 25, 2014; date of current version January 7, 2015. The associate editor coordinating the review of this paper and approving it for publication was G. Y. Lazarou.

The authors are with the Department of Computer Engineering, Sharif University of Technology, Tehran 11365-11155, Iran (e-mail: fattah@ce.sharif.edu; rabiee@sharif.edu; siyari@ce.sharif.edu; a_soltani@ce.sharif.edu; khodadadi@ce.sharif.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LCOMM.2014.2360386

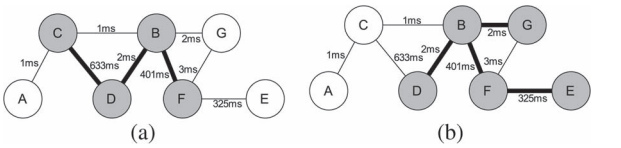

Fig. 1. Two random measurements (bold lines) over a sample network. (a) A measurement from the path $C \rightarrow D \rightarrow B \rightarrow F$. The observed value equals $633 + 2 + 401 = 1036$ ms. (b) A measurement from path $E \rightarrow F \rightarrow B \rightarrow G \rightarrow B \rightarrow D$. $B - G$ has been traversed two times. Hence, the observed value equals $325 + 401 + 2 \times 2 + 2 = 732$ ms.

II. Model and Problem Formulation

We model the network as an undirected graph $G=(V, E)$. The set V contains the graph nodes, representing the hosts/routers of the network which is known, and has cardinality $|V| = n$. Unlike V, the set of edges E, is unknown but still fixed and represents network links. We only consider connected networks in which all nodes are accessible from each other. Based on what we aim to monitor in the network, a scalar is assigned to each node or edge. For example, if we want to identify congested links in a network, we assign a positive integer to each edge, specifying the delay of that link. Fig. 1(a) shows an example network in which the links $C - D$, $B - F$, and $E - F$ are congested. Other links have negligible delays which are considered to be zero in the model.

We assign $x_i \in \mathbb{R}$ to the i-th link in E. Similar to [1], we use *networked data* whenever we refer to the vector $\mathbf{x} = (x_1, x_2, \ldots, x_{|E|})$. Moreover, we suppose vector $\mathbf{x}$ is fixed and k-*sparse*, $\|\mathbf{x}\|_0 \leq k$, which means $\mathbf{x}$ has at most k non-zero elements. It is worth noting that to model real networks, tiny noises or insignificant values must be ignored. Therefore, we can say that the networked data in Fig. 1(a) are 3-sparse. Each measurement is accomplished using a Random Walk (RW). The output value of a measurement is the additive sum of the values x_i corresponding to the visited links. To illustrate the concept, Fig. 1a shows a *ping* measurement, in which an ICMP packet traverses through a path starting from the source host C to the end host F. Here, the corresponding random walk consists of the bold links starting from C to F. Visibly, the observed output value from this measurement is the accumulated delay of the links that are passed through.

To complete the proposed model, let $\mathcal{A}$ be an $m \times |E|$ measurement matrix where each row corresponds to a single measurement, and its entries indicate the number of times a link is included in the corresponding measurement, i.e.,

$$\mathcal{A}_{i,j} = \#\text{ of times link } j \text{ is visited during } i\text{th RW}. \quad (1)$$

We note that previous works [1], [6] assign a random number to $\mathcal{A}_{i,j}$ when link j is visited during measurement i. However, to clearly apply the graph topology constraints, we fill the measurement matrix based on (1).

The observed output values of the measurements construct an observation vector $\mathbf{y}_{m\times 1}$, where $\mathcal{A}\mathbf{x} = \mathbf{y}$. For example, the measurement matrix $\mathcal{A}$ and the observation vector $\mathbf{y}$ obtained from our examples in Fig. 1(a) and (b) are as follows:

$$\mathcal{A} = \begin{array}{c} \begin{matrix} A\text{-}C & C\text{-}B & B\text{-}G & B\text{-}F & B\text{-}D & C\text{-}D & F\text{-}G & F\text{-}E \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \end{array},$$

$$\mathbf{y} = \begin{pmatrix} 1036 \\ 732 \end{pmatrix}$$

CS theory was developed to answer the important question of whether it is possible to find the unknown vector $\mathbf{x}$ from $\mathcal{A}\mathbf{x} = \mathbf{y}$ with only a few number of measurements (i.e., $m \ll n$). Fortunately, for a sparse vector $\mathbf{x}$ and a well-constructed measurement matrix, it is possible to obtain the correct solution with high probability, by using popular sparse recovery techniques [1]–[3].

A. Definitions and Assumptions

Based on previous work on using CS theory to infer and monitor networks [2], [3], we express the following assumptions and definitions.

Definition 1: [3] The undirected graph $G = (V, E)$ is called a (D, c) uniform graph where $D = deg_{min}$, and $c = \frac{deg_{max}}{D}$. For instance, the graph in Fig. 1(a) is (1,4) uniform.

Definition 2: For a standard random walk, initiated from some node u, π_v^* is defined as the stationary probability of being on node v, after a relatively long number of steps, and is calculated as $\pi_v^* = \frac{deg_v}{2|E|}$ [7], where deg_v indicates the degree of node v.

Definition 3: [3] The δ-mixing time of a graph G is the smallest t_0 in which, $\pi_{v,t}$, the probability of being on node v in t-th step, satisfies the following equation:

$$\max\{\forall u \in V, \forall\, t \geq t_0 : |\pi_{t,u} - \pi_u^*|\} \leq \delta.$$

$T(n)$ is defined as the δ-mixing time of G when $\delta = \frac{1}{2cn^2}$.

B. CS over Graphs

As the state-of-the-art in this literature, the authors of [2] have shown that $\mathcal{M} = O(c^4 T^2(n) k \log n)$ independent additive path measurements are sufficient to recover any k-sparse networked data with probability $1 - o(1)$. In their model, each measurement corresponds to a random walk starting from a uniformly random node $v \in V$. It is assumed that a central entity in the network (e.g., the sink in WSNs) will infer the networked data using indirect end-to-end measurements. To this end, the central entity is required to generate m independent random walks. An $m \times |E|$ measurement matrix $\mathcal{A}$ and $m \times 1$ observation vector $\mathbf{y}$ are constructed based on (1), and the output values obtained from those random walks. Using a theorem from [3], it is shown that in graphs with $D \geq O(c^2 k T^2(n))$ and random walk length $\mathcal{L} = O(\frac{nD}{c^3 k T(n)})$, $\mathcal{M}$ random walks are sufficient to recover any networked data with no more than k non-zero elements, over a sufficiently connected network. Clearly, the total number of measurements is much smaller than the network size ($\mathcal{M} \ll n$).

However, considering the case where all the peers require the networked data, the result from [2] is not promising. By assuming that each peer independently generates $\mathcal{M}$ measurements, $n\mathcal{M}$ measurements are generated over the network which may itself cause congestion. In this letter, we propose a new framework in which by using the cooperation among peers, the number of measurements required to recover the networked data is substantially reduced. We also derive an upper bound for the number of required measurements.

III. P2P Compressive Sensing

In this section, we explain our new CS framework which provides the ability for all the peers in the network to efficiently

recover any k-sparse networked data. In this approach, the peers collaborate with each other by using the other peers' measurements. To this end, similar to [6], all peers have their own independent measurement matrix $\mathcal{A}_u, u \in V$. Each peer u not only initiates several random walks, but also uses random walks generated by other peers in order to update its $\mathcal{A}_u$. This cooperation among the peers will sharply reduce the number of required measurements for each peer to recover the networked data, independently. In the proposed model, each peer only knows his neighboring peers and the cardinality of V and E. The topology of the network and the networked data are unknown to all the peers. To start the measurement, peer u is responsible for generating z_u measurements. In contrast to [6], we do not assume that different peers generate an equal number of random walks. This helps minimize the communication overhead imposed by our framework.

Each peer must be able to use incoming random walks to update its measurement matrix $\mathcal{A}$ and observation vector $\mathbf{y}$. Thus, it is necessary that each random walk carries two data types: *a*) **Path**: an array containing the list of peers the random walk has thus far visited, *b*) **Value**: the sum of the data of the links (i.e., x_i components) the random walk has visited. We note that, the same data structures are needed in [2], [6]. This is because it is required in CS theory to know from where a random walk is coming from and what data it observed.

In order to use the mathematical model and the results introduced in [2], we set the length of the random walks $\mathcal{L}' = 2\mathcal{L} = O(\frac{nD}{c^3kT(n)})$. During the measurement process, peer u updates $\mathcal{A}_u$ whenever it receives a random walk with adequate amount of information, which here means that the random walk has at least passed half of its length. This is because young random walks often contain negligible information. The random walk will then continue its path into a random neighboring peer. In addition, peer u informs the random walk's owner, the first peer indicated in the random walk's path, if the received random walk has already completed $\mathcal{L}'$ steps. The owner peer uses these terminated random walks to update its measurement matrix. A peer will stop the measurement process as soon as it receives the terminated message for all of its random walks.

A. The Upper Bound

In this section, we discuss the analytical model for determining an upper bound for the minimum number of required measurements. To this end, we need to make sure that each node can independently recover the networked data with high probability. Therefore, for a peer u, the expected number of measurements in $\mathcal{A}_u$ should meet the $\mathcal{M}$ bound. We use the fact that the probability that a random walk visits node u after step i, where $\frac{\mathcal{L}'}{2} < i < \mathcal{L}'$, is equal to its stationary state probability π_u^* and consequently is independent from its starting peer (i.e., owner). Thus, the expected number of times a random walk visits peer u after $\frac{\mathcal{L}'}{2}$ steps is

$$\sum_{i=\frac{\mathcal{L}'}{2}}^{\mathcal{L}'} \pi_u^* = \mathcal{L}' \frac{deg_u}{4|E|} \tag{2}$$

Therefore, the expected total number of times that peer u updates its measurement matrix from received RWs can be calculated from:

$$\mathcal{V}_u = E\left[\#\text{ of RWs visiting node } u \text{ after } \frac{\mathcal{L}'}{2} \text{ steps}\right]$$

$$= \mathcal{L}' \frac{deg_u}{4|E|} \sum_{v \in V} z_v = O\left(\frac{nD}{c^3kT(n)|E|} deg_u \sum_{v \in V} z_v\right) \tag{3}$$

To ensure that each peer can recover the networked data with high probability, we need to make sure that the number of RWs generated by peer u plus the number of RWs that visit u after $\frac{\mathcal{L}'}{2}$ steps, i.e., $z_u + \mathcal{V}_u$, meets the $\mathcal{M}$ bound:

$$\begin{aligned} z_u + \frac{nD}{c^3kT(n)|E|} deg_u \sum_{v \in V} z_v &= \mathcal{M} \\ \Rightarrow z_u \left(1 + \frac{D^2}{c^3knT(n)} \sum_{v \in V, v \neq u} z_v\right) &\leq \mathcal{M} \\ \Rightarrow z_u = O\left(\frac{\mathcal{M}}{1 + \frac{D^2(\min_{i \in V} z_i)}{c^3kT(n)}}\right) & \end{aligned} \tag{4}$$

where in the first equation, we omitted the $big\ O$ notation, since both sides are order functions. Obviously, from the above upper bound, we can confirm that $z_v < \mathcal{M}$, which means that using the proposed P2P framework, the peers are expected to recover networked data with fewer number of measurements compared to the previous $\mathcal{M}$ bound.

B. Optimization

Inspired by (3) which shows that the expected number of random walks that visit each peer is related to its degree, we optimize the number of required random walks. Our goal is to minimize $\|\mathbf{z}\|_1 = \sum_{u \in V} z_u$, while maintaining the ability to recover networked data in all the peers, i.e.,

$$\begin{aligned} &\arg\min_{\mathbf{z}} \sum_{u \in V} z_u = \|\mathbf{z}\|_1 \\ &\text{s.t. } \forall\, u \in V : z_u \in \mathbb{Z}^+, z_u + \mathcal{V}_u \geq \mathcal{M}, \end{aligned} \tag{5}$$

where from (3), we already have:

$$\mathcal{V}_u = O\left(\frac{nD}{c^3kT(n)} \frac{deg_u}{2|E|}\right) \|\mathbf{z}\|_1 = \mu_u \|\mathbf{z}\|_1 \tag{6}$$

in which μ_u is used to denote $O(\frac{nD}{c^3kT(n)} \frac{deg_u}{2|E|})$. Therefore, we have only one unknown parameter $\mathbf{z}$ in the constraint of (5). Hence, we have:

$$\forall u \in V : z_u \geq \max\{\mathcal{M} - \mu_u \|\mathbf{z}\|_1, 0\} \tag{7}$$

and by summing over all nodes, we conclude that:

$$\|\mathbf{z}\|_1 \geq \sum \left(\mathcal{M} - \mu_u \|\mathbf{z}\|_1\right)^+ \tag{8}$$

where $(x)^+ = \max(x, 0)$.

To minimize the total number of measurements, we must find a value for $\|\mathbf{z}\|_1$ which equates the two sides of (8). To this end, by using the binary search algorithm, and starting from a big positive integer as the first guess, we can easily find the minimum value of $\|\mathbf{z}\|_1$. After finding $\|\mathbf{z}\|_1$, by using the constraint in (7), we can accurately determine the number of random walks each peer must generate. According to (7), after $\|\mathbf{z}\|_1$ and $\mathcal{M}$ bound are found, z_u is inversely proportional to μ_u which itself is directly proportional to $deg(u)$.

From (7), we can observe that only those peers for which $\mathcal{M} - \mu_i \|z\|_1$ is positive are required to generate random walks. Clearly, other peers should not initiate any random walks which will result in a decrease in the total amount of required random walks. Furthermore, peers with smaller μ (i.e., low degree

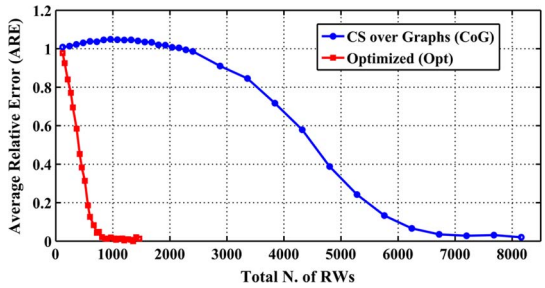

Fig. 2. ARE comparison between *CoG* and *Opt* vs. total number of random walks.

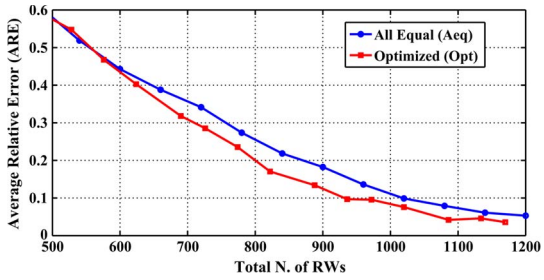

Fig. 3. ARE comparison between *Aeq* and *Opt* vs. total number of random walks.

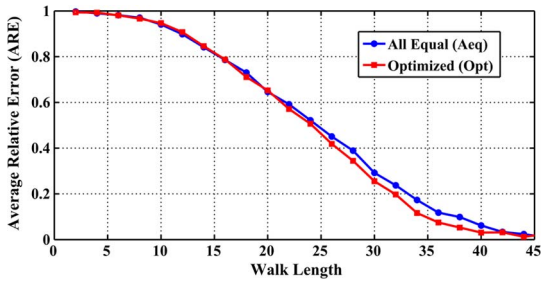

Fig. 4. ARE comparison between *Aeq* and *Opt* vs. random walk length.

peers) must generate more random walks. This matches our intuition that high degree peers should initiate fewer random walks, because they are expected to receive an acceptable number of random walks from other peers.

IV. Results

In this section, we explain the results obtained by running the proposed P2P framework over a Barabasi Albert (BA) random graph. This network was chosen because we know Internet topology follows Power-Laws network models such as BA [8]. We run our simulations over a BA graph with $|V| = 60$ and $|E| = 230$. We compared the recovery performance of three different methods: *a*) **CS Over Graphs** *(CoG)* [2]: all the peers independently perform the state-of-the-art method proposed in [2], *b*) **All Equal** *(Aeq)*: peers cooperate with each other but each generate an equal number of random walks, *c*) **Optimized** *(Opt)*: by solving the optimization (8) and accordingly calculating the vector $\mathbf{z}$ using the constraint of (7), we determine how many random walks each peer must initiate. In these experiments, we set the sparsity to 25 (i.e., we set $x_i = 1$ for 25 random $i \in E$). For fairness, all experiments are repeated 5 times, and the mean values are used in the figures. Moreover, we used new random networked data for each experiment to verify that in our framework, all peers are able to recover any random k-sparse data over the network.

Fig. 2 shows the Average Relative Error *(ARE)* of the recovered networked data in all nodes obtained by Opt and CoG methods. It was seen that, for an equal number of total measurements, the ARE is significantly reduced by Opt. In fact, the ARE for Opt approaches zero while CoG is producing meaningless results (i.e., ARE > 1). For instance, using only 1000 random walks in *Opt* all the peers are able to preciously recover the networked data, while no peer can trust its own recovered data in CoG. Clearly, this huge gap is due to the centralized nature of CoG compared to the collaborative nature of Opt. For a more comprehensive evaluation, we compared the proposed method with Aeq. Fig. 3 illustrates the ARE obtained by Opt and Aeq. The results indicate that for an approximately equal ARE, Opt requires a smaller number of total measurements. This confirms our analysis in Section III, which showed that lower degree nodes must generate more random walks.

Fig. 4 compares the recovery performance of Opt and Aeq as a function of random walk length. The results reveal that for short random walks, the two methods have similar performance. However, as the random walks become larger and high degree nodes observe an adequate number of measurements, our optimized method outperforms Aeq. As expected, for very long random walks, both methods converge to zero ARE.

V. Conclusion

This letter addressed the problem of network monitoring with the assumption that all nodes require access to the entire networked data. By assuming that there are no more than k nonzero data in the network, we used the CS theory to derive an upper bound for the number of required measurements. This upper bound is then optimized in order to determine how many measurements are required to recover the networked data in all nodes with high probability. The results revealed that only low degree nodes are necessary to initiate random walks since high degree nodes use an adequate number of random walks initiated by lower degree nodes.

References

[1] J. Haupt, W. U. Bajwa, M. Rabbat, and R. Nowak, "Compressed sensing for networked data," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 92–101, Mar. 2008.

[2] W. Xu, E. Mallada, and A. Tang, "Compressive sensing over graphs," in *Proc. IEEE INFOCOM*, 2011, pp. 2087–2095.

[3] M. Cheraghchi, A. Karbasi, S. Mohajer, and V. Saligrama, "Graph-constrained group testing," *IEEE Trans. Inf. Theory*, vol. 58, no. 1, pp. 248–262, Jan. 2012.

[4] M. Wang, W. Xu, E. Mallada, and A. Tang, "Sparse recovery with graph constraints: Fundamental limits and measurement construction," in *Proc. IEEE INFOCOM*, 2012, pp. 1871–1879.

[5] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: Recent developments," *Statist. Sci.*, vol. 19, no. 3, pp. 499–517, 2004.

[6] R. Gaeta, M. Grangetto, and M. Sereno, "Local access to sparse and large global information in P2P networks: A case for compressive sensing," in *Proc. IEEE 10th Int. Conf. P2P*, 2010, pp. 1–10.

[7] L. Lovász, "Random walks on graphs: A survey," *Combinatorics, Paul Erdos Eighty*, vol. 2, no. 1, pp. 1–46, 1993.

[8] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power–law relationships of the Internet topology," in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, 1999, vol. 29, no. 4, pp. 251–262.