Local Access to Sparse and Large Global Information in P2P Networks: a Case for Compressive Sensing

Rossano Gaeta, Marco Grangetto, Matteo Sereno
Dipartimento di Informatica, Università degli Studi di Torino, Torino - Italia
Email: {rossano, grangetto, matteo}@di.unito.it

***Abstract*—In this paper we face the following problem: how to provide each peer local access to the full information (not just a summary) that is distributed over all *edges* of an overlay network? How can this be done if local access is performed at a given rate? We focus on *large and sparse* information and we propose to exploit the compressive sensing (CS) theory to efficiently collect and pro-actively disseminate this information across a large overlay network.**

We devise an approach based on random walks (RW) to spread CS random combinations to participants in a random peer-to-peer (P2P) overlay network. CS allows the peer to compress the RW payload in a distributed fashion: given a constraint on the RW size, e.g., the maximum UDP packet payload size, this amounts to being able to distribute larger information and to guarantee that a large fraction of the global information is obtained by each peer. We analyze the performance of the proposed method by means of a simple (yet accurate) analytical model describing the structure of the so called CS sensing matrix in presence of peer dynamics and communication link failures. We validate our model predictions against a simulator of the system at the peer and network level on different models of random overlay networks. The model we developed can be exploited to select the parameters of the RW and the criteria to build the sensing matrix in order to achieve successful information recovery. Finally, a prototype has been developed and deployed over the PlanetLab network to prove the feasibility of the proposed approach in a realistic environment.

Our analysis reveals that the method we propose is feasible, accurate and robust to peer and information dynamics. We also argue that centralized and other distributed approaches, i.e., flooding and gossiping, are unfit in the context we consider.

I. Introduction

Local access to global information is often indispensable in distributed applications exploiting the P2P design paradigm for controlling, monitoring and optimization purposes. Numerous studies have been successful in devising strategies to provide summaries (e.g., averages, ranking, etc.), of some global system property to each peer in an overlay network. In these works it is typically assumed that each peer holds a value, e.g, CPU load, free storage, and that all peers must be provided with an estimate of the summary information computed over the values of the current set of active peers, e.g., [1].

In this paper we consider the following, more difficult, problem: a set V of peers forming a P2P based distributed application organize in a random overlay network by establishing bidirectional connections among them. Each peer maintains a set of c independent information for each outgoing link; the overlay network thus defines c independent global information $x^{(i)}, i = 1 \dots c$ that can be viewed as vectors of as many components as the number of edges ($|E|$) in the overlay. We assume each $x^{(i)}$ to be k-sparse, i.e., $x^{(i)}$ has at most $k \ll |E|$ non-zero elements. Is it possible to provide each peer local access to the full information (not just a summary) $x^{(i)}$? If each peer requires local access at rate λ_r, how can this be done without congesting the overlay network and without exceeding the processing power of each peer?

We develop a technique that exploits both the CS theory [2], [3], [4] and RW to efficiently collect and pro-actively disseminate $x^{(i)}$ across a large overlay network. CS enables compressed acquisition of the information by replacing the standard sample by sample measurement approach with the idea of collecting a (hopefully small) set of *random combinations* of samples. CS theory guarantees that is possible to recover each $x^{(i)}$ from m random projection $y^{(i)} = \Phi x^{(i)}$, where Φ is the $m \times |E|$ CS sensing matrix.

In our technique we use RW with limited lifespan, each one carrying a random combination of the information samples. The CS sensing matrix Φ, obtained according to this technique, is random since the sequence of traversed peers by an RW is random and Φ is sparse since the lifespan of each RW is $\ll |E|$. Furthermore, Φ is the *same for all the c information*.

The access rate λ_r of peers to $x^{(i)}$ does not have impact on the traffic generated. Indeed, each peer generates a fixed number w of RWs during its activity: the higher w the lower the latency. When an RW terminates the peer that first created it is either notified or it timeouts and a new RW is generated. Therefore the total amount of RWs in the overlay is bounded by $w \cdot |V|$, independent from λ_r. Furthermore, since we exploit RWs our technique is resilient to peer churning and unreliable message transmission.

Since CS allows peers to compress the RW payload in a distributed fashion, given a constraint on the RW size, e.g., the maximum UDP packet size, this amounts to being able to distribute larger information and to guarantee that a large fraction of the global information is obtained by each peer.

We analyze the performance of the proposed method by means of a simple (yet accurate) analytical model describing the structure of Φ in presence of peer dynamics and communication link failures. We validate our model predictions against a simulator of the system at the peer and network level on different models of random overlay networks. The model we developed can be exploited to select the parameters of the RW and the criteria to build the sensing matrix in order to achieve successful information recovery. Finally, a prototype has been deployed over the PlanetLab network to prove the feasibility of the proposed approach in a realistic environment.

Work exploiting CS in computer science are discussed in Section II. The reference system and the RW based CS technique are presented in Section III. The first step in our analysis is the investigation of the CS recovery probability in the case where the Φ matrix is obtained by adding random binary sparse rows. In Section IV we observe a dependence of the recovery probability on the average number of non-zero entries in *columns* of Φ; in particular, when this number is greater than a threshold information recovery can be achieved with probability 1 for a given k. The technique is subsequently analyzed by means of a simple (yet accurate) analytical approximation of the probability distribution of the number of non-zero entries in columns of Φ that we develop in Section V. The analytical model includes possible unreliable communication and peer dynamics that can join and leave the overlay network alternating between active and idle periods. The model is validated against results obtained from a system simulator on several different types of random networks in Section VI; this section also describes results we obtained from a prototype implementation we deployed and tested on PlanetLab. Section VII compares our technique with other centralized and distributed approaches arguing that they are unfit in the context we consider. Finally, Section VIII draws conclusions and outlines some lines of future research.

II. Related work

Some applications of the CS theory in the field of measurements in computer and sensor networks have recently appeared. Identification of significant patterns in network traffic has been the subject of [5], while CS is used to reduce the memory cost of per-flow measurements in routers and switches in [6]. Very recently, [7] exploited CS theory to define an interpolation technique to reconstruct missing values in traffic matrices based on direct and indirect measurements.

In the sensor networks domain, in [8], [9] CS is used in wireless sensor networks to design distributed acquisition and detection algorithms. In this context one assumes that each sensor knows a sample of the signal x and random sketches at a collecting peer can be obtained by exploiting the interference on the wireless communication channel.

The works in [10], [11] deal with the more general scenario of decentralized acquisition and compression for networked data and share some resemblance with our work. In these papers the sensing peers form a multi-hop network and one cannot rely on interference to form random combinations of the signal. Each peer is assumed to hold a sample of the signal and is assumed to use random coefficients to build a measure. A consensus technique based on random gossiping is proposed to create and disseminate the measurement vector to all peers. On random geometric graphs the number of single hop communications is $\Theta(mcn^2)$. The major limitation of this approach is that the number of sensing peers must be known to all participants to initialize the consensus algorithm. As a consequence, only static networks are considered. Furthermore, the application of the same technique to recover a set of c signals defined on the network *links* would dramatically increase its communication complexity. Last but not least, these strategies may become unfeasible as the access rate λ_r of peers increases as discussed in Section VII.

III. The technique

In this section we provide a brief summary of the main issues of CS theory we exploit in this paper. We refer the reader to [2], [3], [4], [12], [13], [14] for a detailed treatment of this subject. We also define the reference system and the technique we developed.

A. CS main facts

CS enables compressed acquisition of information by replacing the standard sample by sample measurement approach with the idea of collecting a small set of *random combinations* of samples. CS theory guarantees that is possible to recover each $x^{(i)}$ from m random combination $y^{(i)} = \Phi x^{(i)}$, where Φ is the $m \times |E|$ CS sensing matrix and $m \geq \alpha k \log |E|$, where α is a constant. In particular, it has been proved that the probability of information recovery depends on the so called *restricted isometry property* (RIP), which requires that every set of k (or less) columns of Φ forms an approximatively orthonormal basis. In other words, this assures that k-sparse information does not fall into the null space of Φ. If Φ satisfies the RIP property then $x^{(i)}$ can be recovered with probability 1 in a certain range of k and m. In this setting, the recovery algorithm for $x^{(i)}$ can be recast as the following linear program: $\min ||\hat{x^{(i)}}||_1$ sub. to $\Phi \hat{x^{(i)}} = y$, where $\hat{x^{(i)}}$ represents the recovery of $x^{(i)}$.

In theory, Φ shall be constructed according to some criterion guaranteeing the RIP property. The performance of binary sparse random matrices has been recently studied in [12], [13], [14]. The most important result is that a sparse Φ permits to simplify the updating and recovery process without impacting on the CS performance. In particular, in [13] it is shown that a RIP-1 property can be used as guarantee for CS recovery and that, more importantly, binary sparse matrices constructed by placing $d \ll m$ 1's in d random positions of each column yield optimal recovery. The parameter d is usually termed as the *column degree* and it was noted in [12] that any value $8 \leq d \ll m$ yields almost the same recovery probability.

B. The reference system

Let us consider a set of peers organized in an overlay network. We denote the set of peers as V and the set of logical connections among them as $E \subseteq V \times V$. Connections are assumed to be bidirectional. The neighborhood of peer v is defined as $N(v) = \{u \in V : (v, u) \in E\}$. We assume any random connected overlay: we make no assumptions on the overlay formation algorithm.

Each peer stores a set of c independent information for each outgoing link; to improve readability and to avoid cluttering the notation in the following we consider $c = 1$ and consequently drop the dependency on information component i. We consider a global information x defined on the edges of the overlay network, i.e., x is a $|E|-$dimensional vector. We denote the value of the information for a generic edge e as x_e. The information we consider is k-sparse meaning that each sample is non-zero with probability q such that $q \cdot |E| = k$. Each peer is interested in recovering x from m random combinations $y = \Phi x$; to this end, each peer stores its own Φ that is the $m \times |E|$ random sparse binary CS matrix and its own m vector y.

C. The proposed technique

To gather random combinations of x each peer generates w RWs, i.e., messages that are forwarded to a randomly chosen neighbor. The total number of RWs is thus equal to $w \cdot |V|$. Each RW r is allowed to be forwarded for a maximum number of hops denoted as TTL. The TTL value is the same for all RWs. The peer that receives an expired RW notifies the peer that has originated the RW; the RW is then regenerated in order to keep a constant number of RWs visiting the overlay at any time. A RW contains several pieces of information: the identities of the visited peers (hence the identities of the traversed edges), the accumulated value of the random combination, and the residual number of allowed hops. Each time a RW is forwarded the identity of the receiving peer is added, the combination is updated, and the residual number of allowed hops is decreased by one unit.

A peer v updates the combination of an RW r as follows: v extracts the accumulated combination value y_r and updates it by summing its contribution, i.e., $y_r = y_r + \Phi_{r,e} \cdot x_e$ where e is a randomly chosen outgoing edge and $\Phi_{r,e}$ represents an element of the CS matrix. Each element in Φ can be chosen in several ways: the simplest choice corresponds to set $\Phi_{r,e} = 1$ yielding a binary CS matrix. In general, $\Phi_{r,e}$ can be randomly chosen using a given probability distribution. In our system, a peer that receives an RW sets $\Phi_{r,e} = \pm 1$ with uniform probability. The explicit values of $\Phi_{r,e}$ need not to be stored in the RW provided that any peer is able to repeat the random generation process. For instance, the seed of the random number generator used by v can be initialized to a value that can be reconstructed from the identity of e and from the position of e in the RW performed by r.

A peer that receives an RW can use the information it carries to add a row in its CS matrix Φ. This row contains a non-zero entry only for the links visited by the RW. Furthermore, the peer inserts the combination of x computed on the visited links in its y vector. If the number of hops taken by the RW (that we denote as $s(r)$) is much smaller than $|E|$ then Φ admits an efficient sparse representation. Furthermore, a sparse representation of Φ is unavoidable since E, and thus its cardinality, is not known in advance to peers. A peer adds rows to its own Φ until m RWs are received; when Φ rows are all filled a new row overwrites the oldest one so to realize a sliding window mechanism to store only the m most recent pieces of information. The decision to store the information carried by a received RW r is taken by peers using a probability $p_s(r)$ that depends only on the number of hops already taken by r, i.e., on the amount of information carried by the combination. For instance a peer can insert in Φ the information carried by r if $s(r) \geq s_m$, where the minimum number of hops s_m is a parameter of the technique.

Communication is assumed to be unreliable, i.e., RWs may be lost with probability p_{loss} upon each transmission. No loss detection and recovery mechanism are employed by peers that alternate between active and idle periods. An active peer performs all operations we described while an idle peer leaves the overlay without notifying its neighbors about its departure, i.e., departures are all silent. If a peer selects an idle neighbor peer the transmitted RW gets lost; peers employ a timeout mechanism to detect such losses and to regenerate RWs. An idle peer becoming active maintains its neighbors and discards previous information stored in its own Φ and y data structures. It also generate its w RWs to contribute to the spreading of random combinations of x. The information x is constituted by associating a random integer value x_e to each outgoing edge with probability q.

Time is assumed to be slotted; at each time slot an active peer manages RWs as described. Furthermore, with probability p_{idle} a peer switches to the idle state departing from the overlay. Similarly, with probability p_{active} an idle peer rejoins the overlay network and discards all rows in Φ accumulated during the previous active period. This means that each time a peer activates it has to wait for a startup time necessary to fill all m rows in Φ. Clearly, the average activity period must be greater than the average startup time to allow for information recovery. To help the reader Table I summarizes the notation that will be used throughout the rest of the paper.

Remarks

The scenario we consider is the most unfavorable because communications are unreliable and losses are neither detected nor recovered. Departures are not notified to neighbors therefore RWs that are forwarded to idle peers are lost. Last but not least, each time a peer rejoins the overlay network it starts anew by discarding all rows of Φ accumulated during its last active period.

IV. CS MATRIX PROPERTIES AND PERFORMANCE

The performance of a CS system depends on the properties of matrix Φ. In theory, Φ shall be constructed according to some criterion guaranteeing the RIP property. The technique we propose yields random, sparse and binary matrix Φ at each peer.

These matrices Φ are constructed by rows and we cannot enforce any guarantee on the column degree as discussed in Section III-A. In the following we empirically analyze the CS performance when Φ is built by rows. To this end let us consider a binary signed sparse Φ whose rows are constructed as follows: first pick up a random integer g in the range

Symbol	Description
V	set of peers
E	set of connections
u, v	generic peers in V
e	generic edge in E
x	information to recover
x_e	information value for edge e
$N(v)$	neighborhood of peer v
q	probability of non-zero sample in x
m	number of random combinations of x
y	vector of m random combinations of x
Φ	$m \times \|E\|$ random binary CS matrix
$d(\overline{d})$	Φ column degree (and its average)
r	generic RW
w	number of RWs per peer
TTL	maximum number of hops for RW
$s(r)$	number of hops taken by RW r
$p_s(r)$	probability to store information of RW r
s_m	minimum number of hops before RW insertion in Φ
p_{loss}	probability of lossy transmission
p_{idle}	probability an active peer goes idle
p_{active}	probability an idle peer goes active

TABLE I
SYMBOL NOTATION AND DESCRIPTION.

$[s_m, TTL]$, then select g random integers in the range $[1, |E|]$ and place ±1 in the corresponding column positions. The proposed construction method assumes that every outgoing link is visited with the same probability by every RW and that every peer v stores a set of m RWs that have performed at least s_m hops. Moreover, we assume that the length of the stored RWs is uniformly distributed between s_m and TTL.

Clearly, the random construction of Φ imposes a row degree distribution that depends on parameters TTL and s_m. As a consequence, the column degree distribution depends on TTL, s_m and the number of combinations m. In the following we estimate $\overline{d}(TTL, s_m, m)$, which is the average column degree yielded by the proposed construction method.

To test the performance of the proposed RW based Φ construction method we use an experimental setup similar to the one reported in [13], where binary matrix with fixed column degree are studied. We randomly generate the k-sparse information x with $|E| = 1000$ samples; $k = q|E|$ out of the $|E|$ samples are set to positive random integer values. We fix the values for q, TTL and s_m and let m vary. We then estimate the empirical recovery probability estimated from 100 CS recovery trials for each point (q, m) on a discrete grid, using 100 independent random samples of the pair x,Φ. In Figure 1(a) we show the obtained CS recovery region as a function of q and m for some values of the parameters TTL and s_m. Each curve represents the lowest value of m yielding an empirical recovery probability equal to 1, i.e., 100 independent successful CS reconstructions have been reported for a given q, m point on the curve. Therefore, all points lying below each of the reported curves represent the region where CS yields a perfect reconstruction of x. It is worth noting that different choices of TTL and s_m yield very similar performance, i.e. for a given q they require almost the same number of combinations. In Figure 1(b) the same results are shown by substituting the value of m on the x-axis with the corresponding value of $\overline{d}(TTL, s_m, m)$. Figure 1(b) clearly shows that CS recovery is not possible if $\overline{d} < 8$. This finding is in line with observation made in [12], [13], when replacing d with $\overline{d}$. This is a key observation for the selection of the parameters of our CS system. As an example, one may want to fix the number of combinations m that a peer can store, i.e., the memory requirements, and select the minimum TTL and s_m values guaranteeing information recovery for a certain sparsity q by enforcing a constraint on $\overline{d}$.

V. MODEL DEVELOPMENT

In this section we describe the analytical model we developed to characterize the structure of the CS matrix Φ at each peer. In particular, we derive the probability distribution of the number of non-zero entries for columns of Φ, and an approximate formula to compute the average filling rate of Φ that can be used to obtain the average startup time.

A. The model

Consider an RW r that reaches peer v after l hops (that is, $s(r) = l$ and $1 \leq l \leq TTL$) and a randomly selected edge $e \in E$. The model we develop is based on the following approximation: the average probability that e has been traversed by r is equal to $\frac{l}{|E|}$, i.e., we assume that an RW of length l is equivalent to a random sampling of a set of l edges in E. Under this assumption the probability that edge e is visited h times by k RWs reaching v after l hops is simply $B(k, \frac{l}{|E|}, h)$, i.e., a binomial probability distribution whose population is equal to k with parameter $\frac{l}{|E|}$. Since each RW is inserted in Φ with probability $p_{s(r)}$ the probability distribution $\{d_h\}_{h=0}^{m}$ of the number of rows where the entries corresponding to edge e are non-zero is given by

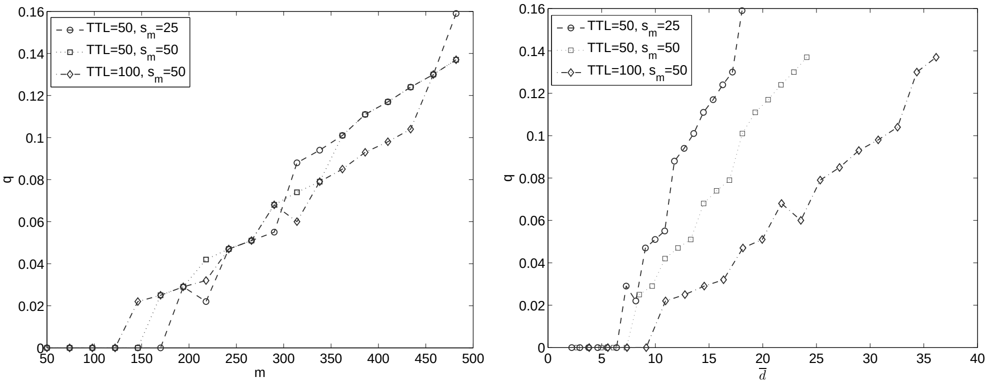
$$d_h = R(m, |E|, TTL, h) = \sum_{l=1}^{TTL} f_l \cdot B(m, \frac{l}{|E|}, h) \qquad (1)$$

where $f_l = \frac{p_l}{\sum_{l=1}^{TTL} p_l}$ hence $\forall l, 0 \leq f_l \leq 1$ and $\sum_{l=1}^{TTL} f_l = 1$. From this probability distributions the fraction of x that *cannot* be reconstructed by peers is represented by d_0.

The main index we are interested in is

$$\overline{d} = \frac{\sum_{h=1}^{m} h \cdot d_h}{1 - d_0} = \frac{m}{|E|} \cdot \sum_{l=1}^{TTL} f_l \cdot \frac{l}{1 - (1 - \frac{l}{|E|})^m} \qquad (2)$$

representing the average value of the number of non-zero entries per column in Φ; it is the first moment of the conditional distribution $R(m, |E|, TTL, h|h > 0)$ and it determines the feasibility of the information reconstruction as discussed in Section IV since it determines the probability of successful recovery for x. It is easy to observe that $\overline{d}$ depends on the length of the RWs that are inserted in Φ: the higher the TTL the higher $\overline{d}$. Since in our system a peer could insert a random combination in Φ only if at least s_m hops have been taken, it follows that the probability distribution in Equation (1) turns



(a)

(b)

Fig. 1. CS recovery region with RW in the case $|E| = 1000$ as a function of m, q (a) and $\overline{d}, q$ (b).

to be $p_l = 0$, if $1 \leq l < s_m$ and 1 if $s_m \leq l \leq TTL$. Thus we obtain

$$d_h = \frac{\sum_{l=s_m}^{TTL} B(m, \frac{l}{|E|}, h)}{TTL - s_m + 1}. \quad (3)$$

The system we consider does not provide reliable communication therefore RWs might get lost during transmissions. In this case, the model in Equation (1) can be adapted to consider losses at each transmission with probability p_{loss} by setting $\forall l, p_l = (1 - p_{loss})^l$, i.e., an RW is inserted in Φ with probability equal to l consecutive error-free transmissions. Clearly, the model that includes unreliable transmissions can be combined with the model that selects the minimum length of an RW to obtain

$$d_h = \frac{\sum_{l=s_m}^{TTL} (1 - p_{loss})^l \cdot B(m, \frac{l}{|E|}, h)}{\sum_{l=s_m}^{TTL} (1 - p_{loss})^l}. \quad (4)$$

Another feature of the system we consider is the possibility of peers to join and leave the overlay network. Active peers go into idle state with probability p_{idle} while idle peers activate with probability p_{active}. It follows that the average probability of finding a peer in the idle state is given by $p_{off} = \frac{p_{idle}}{p_{active} + p_{idle}}$ while the average probability of finding a peer in the active state is given by $p_{on} = 1 - p_{off}$. The general model defined in Equation (1) can be tailored to cope with peers dynamics by setting $\forall l, p_l = [1 - (p_{off} + p_{on} \cdot p_{idle})]^l$, i.e., an RW is inserted in Φ with probability equal to l consecutive choices of active neighbors that do not switch to the idle state. Once again, a model incorporating both RW selection and peer dynamics is obtained as

$$d_h = \frac{\sum_{l=s_m}^{TTL} [1 - (p_{off} + p_{on} \cdot p_{idle})]^l \cdot B(m, \frac{l}{|E|}, h)}{\sum_{l=s_m}^{TTL} [1 - (p_{off} + p_{on} \cdot p_{idle})]^l}. \quad (5)$$

Finally, a complete model including RW selection, unreliable transmission, and peer dynamics can be defined as

$$d_h = \frac{\sum_{l=s_m}^{TTL} (1 - p_{fail})^l \cdot B(m, \frac{l}{|E|}, h)}{\sum_{l=s_m}^{TTL} (1 - p_{fail})^l}. \quad (6)$$

where $p_{fail} = p_{off} + p_{on} \cdot [1 - (1 - p_{loss}) \cdot (1 - p_{idle})]$

According to the system described in Section III, each time a peer activates and rejoins the overlay network it discards previously filled rows of Φ. It is then important to characterize the startup delay of each peer defined as the time before all m rows of Φ are filled. To this end, we approximate the probability that an RW starting at peer s is at peer v after t hops as $\frac{|N(v)|}{|E|}$, i.e., the limiting value for $t \to \infty$. This approximation is certainly accurate when t is greater than the mixing time of the RW on the overlay network. We introduce a further approximation by setting this probability equal to $\frac{1}{|V|}$, i.e., the probability an RW is at peer v is a uniform probability. This second approximation is expected to be acceptable for random overlay networks where the degree distribution of peers is peaked around its average value. Under these hypothesis the average number of RWs that is received by a peer in a time slot (the filling rate of Φ) is simply given by $r_f = w \cdot |V| \cdot \frac{1}{|V|} = w$. In the general case

$$r_f = w \cdot \frac{\sum_{l=s_m}^{TTL} (1 - p_{fail})^l}{TTL}, \quad (7)$$

therefore the startup delay is simply obtained by $T_f = \frac{m}{r_f}$.

VI. Results

In this section we describe the simulator we developed to validate the analytical model presented in Section V; in particular, the accuracy of Equations (2) and (7) is validated. Furthermore, we describe the implementation of the proposed CS technique in a prototype that has been deployed and tested on PlanetLab to demonstrate the feasibility in a real distributed network of planetary scale.

A. The simulator

The performance of the proposed system has been analyzed by means of a simulator. The simulator, developed in C++ language, works at the overlay level managing logical connections E among the peer set V. The overlay network is one of the inputs of the simulator in the form of graph instance produced using the igraph C library. Time is assumed to be slotted. Two kinds of peer are implemented.

The forwarding peer is able to update and propagate (with probability $1-p_{loss}$) the RWs received in the previous time slot. Timeouts are scheduled in order to detect RW losses and regenerate them. The update procedure is performed by using $\Phi_{r,e} = \pm 1$ with uniform probability. Therefore, the combinations are updated by summing/subtracting a given sample x_e. At each time slot the peer state (active or idle) is updated according to the probabilities p_{idle}, p_{active}. A peer switching from the idle to the active state updates the information on its outgoing edges with probability p_{update}.

The second peer class is represented by the sensing peer, that inherits all the forwarding peer data structures and methods and adds the CS functionalities. The sensing peer maintains a circular buffer of size m, where the most recent RWs, meeting the requirements on the minimum number of hops s_m, are stored. This data structure permits to extract the CS matrix Φ: from m RW payloads the set of edges $E_v \subseteq E$ observed by v is extracted. Each RW is mapped onto a row of Φ by regenerating the corresponding coefficients $\Phi_{r,e}$. CS recovery is attempted as soon as Φ fills up for the first time. Then, more attempts are performed when a certain percentage of RWs, e.g. 10%, has been refreshed in the circular buffer. The CS recovery is based on the algorithm in [15] (implemented using the LAPACK and sparse BLAS libraries), that has been selected because of its limited computational cost, compared to the linear programming approaches. The research in the area of CS recovery algorithms is very active and our choice is not meant to be the optimal one. The goal of our implementation is to demonstrate that the proposed system is feasible also from the point of view of the computational cost.

B. Simulation results

In this section we first present the validation results we obtained by comparing the accuracy of the model predictions against detailed simulation of the RW based CS on random networks with $|V| = 2000$ peers and average number of outgoing connections $z = 10, 20, 30$. We considered instances of synthetic random graphs using two different models: Erdös-Rényi graphs with probability of an edge between any two peers equal to $\frac{z}{|V|}$ and random graphs whose degree distribution is uniform in the interval $[z-5, z+5]$. [1] We also considered the following system parameters: $TTL = 50$, $s_m = 30$, $w = 1$, $p_{idle} = 5 \cdot 10^{-4}$, $p_{active} = 9.5 \cdot 10^{-3}$ (hence $p_{off} = 0.05$), $p_{loss} = 0.001$, information sparsity $q = 0.1$, and increasing values for m in the range $[500, 2500]$. The simulation has been run until 10% of randomly chosen peers in the network filled their Φ matrix. This limitation is due to the impossibility of storing Φ and y for all peers in the RAM of the workstation we used. This limitation affects all the simulator results, whereas it will be removed when using the prototype implementation on PlanetLab. For each simulation on a graph instance results have been averaged over all sensing peers. Each point in the following curves has been obtained by considering 30 instances of each type of random graphs and results are averaged over all graph instances.

TABLE II
CS RECOVERY FOR $TTL = 100$, $s_m = 75$ VERSUS q.

q	p_{CS}	$n_f/\|V\|$	d_0
	Uniform		
0.010	0.990	0.0	0.000073
0.015	1.0	0.0	0.000070
0.020	1.0	0.0	0.000075
0.025	0.431	0.15	0.000067
	Erdös-Rényi		
0.010	1.0	0.0	0.000070
0.015	0.990	0.0	0.000063
0.020	1.0	0.0	0.000078
0.025	0.384	0.16	0.000066

The results of this validation are presented in Figure 2 where the relative error $|\frac{\hat{d}-\overline{d}}{\overline{d}}|$ is plotted for increasing values of m where $\hat{d}$ is the estimated average number of non-zero entries in columns of Φ as computed by the simulator. It can be noted that the predictions of Equation (2) are very accurate for both types of graphs. In all cases the relative error approaches 0 as m increases.

Figure 3 presents the validation for the filling rate r_f as defined in Equation (7) in a setting where $z = 10$, $TTL = 100$, $s_m = 1$, $p_{idle} = 10^{-3}$, and $p_{loss} = 10^{-2}$ for increasing values of p_{off}. Also in this case we note that the model is very accurate despite several approximations we introduced; it is able to represent the behavior of the simulated $\hat{r_f}$ rather closely for all the considered network models.

We conclude that the analytical model we developed is accurate and can safely be exploited for system design and optimization.Due to the lack of space we only reported a subset of the validation results we obtained for several combinations of the system parameter values.

The simulator has been used to test performance of the proposed technique in a controlled scenario. We consider a static network as in the case of the model validation with $p_{off} = p_{loss} = 0$, $w = 1$, $|V| = 2000$ and $z = 10$. As already noted, CS recovery performance depends on the value of $\overline{d}$ as described in Section V. We fix the number of combinations that can be stored by each peer $m = 2500$ and select the RW parameters $TTL = 100$ and $s_m = 75$ imposing $\overline{d} > 10$ according to the model (2), where $|E| = |V| \cdot z$. In Table II we show the experimental results as a function of the information sparsity q. All results have been averaged over a subset of 100 sensing peers, that are allowed to perform 5 CS recovery attempts. The performance is measured in terms of the average CS recovery probability p_{CS} and the fraction of peers that always fail CS recovery $n_f/|V|$. Moreover, the experimental value of the fraction of the missed edges d_0 is reported. The obtained results confirm that the model (2) predicts the behavior of the CS matrix so as to guarantee a

[1] We also validated the model on Watts-Strogatz small world graphs with z connections and where 10% of links are randomly rewired and on regular random graphs with z edges per peer. Results accuracy is comparable to the presented cases and are omitted due to the lack of space.

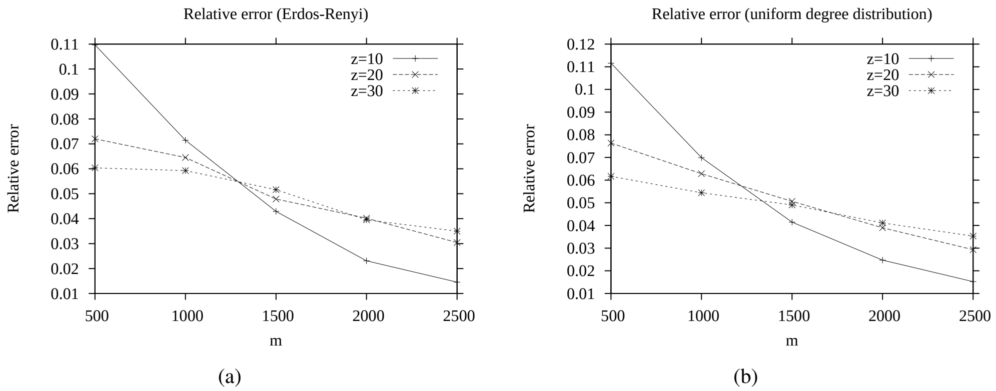


Fig. 2. Relative error $|\frac{\hat{d}-\bar{d}}{d}|$ as a function of m for Erdös-Rényi (a) and uniform (b) random graphs.

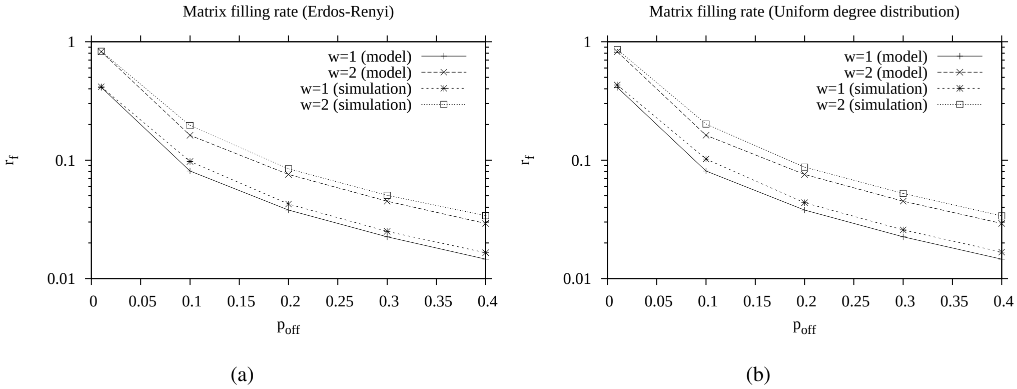


Fig. 3. Filling rate r_f as a function of p_{off} for Erdös-Rényi (a) and uniform (b) random graphs.

successful information recovery in a proper range of sparsity q. Our experimental results show that is possible to reliably reconstruct x up to $q = 0.02$, i.e. $k = 400$ non-zero values over $|E| = 20000$ samples from $m = 2500$ random combinations. The simulator can be used to test the proposed technique in presence of both peer and information dynamics. In this case we consider $p_{loss} = 0$, $p_{idle} = 10^{-5}$ and $p_{off} = 0.02$ using $m = 2500$, $TTL = 150$ and $s_m = 55$. In Table III the experimental results are reported as a function of p_{update} for a fixed value of sparsity $q = 0.01$. In order to appreciate the effect of the time varying information on the CS recovery we show, along with p_{CS} and $n_f/|V|$, the ratio of the reconstructed edges which are active in the instant when a peer performs the signal recovery (active ratio). The performance reported in Table III shows that is possible to design the CS system so as to cope with both peer and signal dynamic; of course, p_{CS} decreases and the number of peers that failed recovery increases as p_{update} gets larger.

C. PlanetLab prototype

The proposed technique has been implemented in a prototype that has been deployed and tested on PlanetLab. PlanetLab allowed us to demonstrate the feasibility of the proposed approach in a real distributed network of planetary scale. Moreover, in this case we allow peers to dynamically update the information associated with the outgoing links; in particular information changes each time a new peer joins the overlay.

TABLE III
CS RECOVERY FOR $TTL = 100$, $s_m = 55$, $q = 0.01$ AND $p_{off} = 0.02$ VERSUS p_{update}.

p_{update}	p_{CS}	$n_f/\lvert V\rvert$	d_0	active ratio
		Uniform		
0.008	0.909	0.0	0.001	0.967
0.01	0.667	0.04	0.001	0.962
0.02	0.437	0.16	0.001	0.962
		Erdös-Rényi		
0.008	0.881	0.01	0.002	0.965
0.01	0.483	0.11	0.002	0.961
0.02	0.349	0.21	0.001	0.962

The prototype implements two functionalities, namely the creation and maintenance of a random overlay network of peers and the distribution of the RWs to be used to recover the information samples. Only UDP datagram communications are used among peers. A random mesh overlay is built using a rendez-vous peer, i.e. the *tracker*, that stores the list of the participating peers and provides a random subset of such list upon request. A new joining peer retrieves a set of peers from the tracker, issues a connection request to them and forms its neighborhood with the ones that reply positively.

Each peer keeps a number of neighbors between $z_{\min}$ and $z_{\max}$. In the present implementation we set $z_{\min} = 15$, $z_{\max} = 25$. The sparse information x is created dynamically; each peer associates an integer value to every outgoing edge as soon as it is able to finalize a connection request. The corresponding sample x_e is set to a random positive integer with probability q, to 0 otherwise. When a peer leaves the network its neighborhood and the tracker are informed so as to update their lists. Both the peers and the tracker use timeouts to infer silent departures, e.g. because of peer crashes. The proposed RW based system is rather simple to implement. At startup, each peer initiates w RWs. The peer is authorized to initiate a new RW in two cases: one of its RWs has performed TTL hops in the network or a timeout T_r has expired. The first event is reported by the peer that has collected the RW after TTL hops. Since all the communications are based on UDP the timeout T_r is crucial to regenerate an RW in presence of packet losses or peer failures. The following experiments have been worked out with $T_r = 30 \cdot (TTL+1)$ ms. Every received RW is updated and propagated to a neighbor. When an RW expires in a peer its originator is signaled. Each peer uses a circular buffer of size m to store the most recent RWs that have already performed at least s_m hops. Each peer attempts CS reconstruction as soon as the circular buffer fills up or at least 10% of the collected RWs have been refreshed.

The described prototype has been deployed on PlanetLab and tested under 3 scenarios with different behaviors of the peers. In the first case we emulate a stable network where 440 peers stay connected to the overlay for 10 minutes. The number of peers is determined by the resources available on PlanetLab during the experiments. In the second scenario we consider an overlay where peers join the overlay at a pace of 10 peers per second (up to the limit of 440 peers), then remain in the overlay for 10 minutes. In the third scenario we introduce peer churn, letting the peer join and stay in the overlay for random exponential time intervals. In this case the available 440 peers cyclically join the overlay for an average time of $T_{on} = 900$ s and turn off for an average time of $T_{off} = 30$ s. In this latter case the performance has been measured on period of 30 minutes after the overlay reaches the steady state population, i.e., the average value of connected peers oscillates around the theoretical value $440 \cdot \frac{T_{on}}{T_{on}+T_{off}}$. Although the churn may appear very limited, it should be noted that each time a peer joins the network it gets new neighbors from the tracker and updates the information, thus making the CS reconstruction a challenging issue. In all the scenarios we considered a sparse information with $q = 0.01$ and we use $w = 20$ RWs per peer.

Each peers v logs the values of the created edges and all the links values obtained at every CS recovery attempt. This allowed us to evaluate the following performance indexes: the number of links $|E_v|$ observed by v, the reconstruction error rate RE$_v$ for each CS recovery attempt, defined as the ratio between the number of reconstruction errors and the number of observed links $|E_v|$. We compute also the percentage PR$_v$(tol) of recovery attempts with an error rate below a tolerance tol, to measure the reliability of collected information, given a certain admissible error margin. All previous indexes can be averaged over all peers to get global performance indexes.

TABLE IV
AVERAGE RECONSTRUCTION ERROR RATE (RE) AND AVERAGE PERCENTAGE OF RECONSTRUCTION BELOW TOLERANCE 10^{-3} (PR).

Scenario	m	TTL	s_m	RE	$PR(10^{-3})$
Stable	1500	150	45	$9.2 \cdot 10^{-5}$	98.0
	3000	50	30	$1.5 \cdot 10^{-4}$	99.1
Mass arrival	1500	150	45	$4.6 \cdot 10^{-4}$	92.3
	3000	50	30	$2.8 \cdot 10^{-4}$	95.9
Churn	1500	150	45	$2.1 \cdot 10^{-3}$	20.6
	1500	100	30	$1.5 \cdot 10^{-3}$	43.5
	2500	50	30	$1.2 \cdot 10^{-3}$	56.2

In Table IV the performance indexes, averaged over all the peers, are shown for the 3 different scenarios. The parameters of the RW, namely m, TTL and s_m have been selected using the analytical model, i.e., imposing $\overline{d} > 10$. It can be noted that our technique yields an accurate estimate of the information in all the considered scenarios. As obvious, the stable overlay represents the most favorable scenario. With the mass arrivals peers are still very likely to reliably recover the information. In the most dynamic case, i.e. peer churning and updating the information, the performance of the technique improves by decreasing the value of TTL. This can be explained by noting that a reduced TTL limits the effect of RWs spreading information on links that do no exist any more because of peer departure. In this case the column degree corresponding to such links can only decrease with time contributing to reduce the value $\overline{d}$.

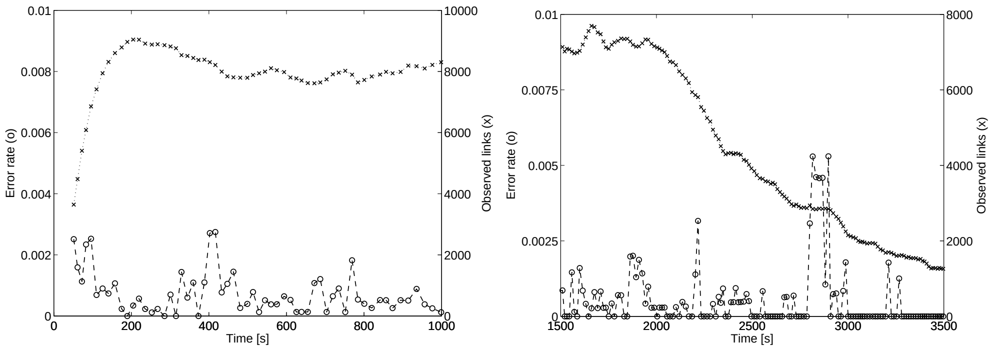
In the churn case, the average results in Table IV are completed with the empirical cumulative distribution function of the RE shown in Figure 4. It can be noted that 90% of the CS reconstructions yield an error rate below 0.002, that is a good result in such a difficult setting. Finally, in Figure 5 the temporal behavior of $|E_v|$ and RE_v for two sample peers is shown before and after the network reaches its steady state. It can be noted that, even in this case, the technique is able to track the number of links in the system with a low RE both when the overlay is forming (a) or most of the peers are leaving (b).

VII. Comparison with other approaches

In this paper we discussed the feasibility of CS based techniques to allow all peers to have local access to large and sparse global information defined on edges of a random overlay networks. In this section we argue that approaches that do not exploit compression are less efficient. In the sequel we denote as λ_r the rate of global data access request of a peer and μ the service capacity of peers.

A. Centralized solution

A centralized solution can be conceived where all peers periodically pack the non-zero $x^{(i)}, i = 1 \ldots c$ data for all their outgoing links into a data packet that is sent as an update to a common data repository that could be implemented in



(a)

(b)

Fig. 5. Reconstruction error rate R_v as function of time (cross marker) and corresponding E_v for two sample peers in presence of churn.

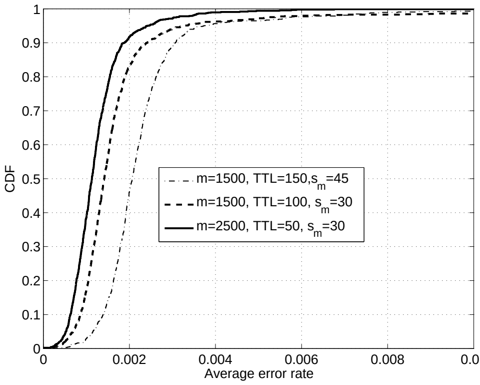


Fig. 4. Cumulative distribution function of the reconstruction rate in presence of churn.

the application rendez-vous point, e.g., the BitTorrent tracker. Peers access the global data by sending request messages to the rendez-vous point that provides response messages. Unfortunately, this straightforward solution suffers from poor scalability and resilience since the rendez-vous point communication and processing resources could be easily saturated. In fact, if we neglect the load offered by the update messages, the load factor on the rendez-vous point is equal to $\frac{|V|\cdot\lambda_r}{\mu}$ that quickly becomes greater than 1 leading to requests loss and unbounded delays.

B. Decentralized solutions

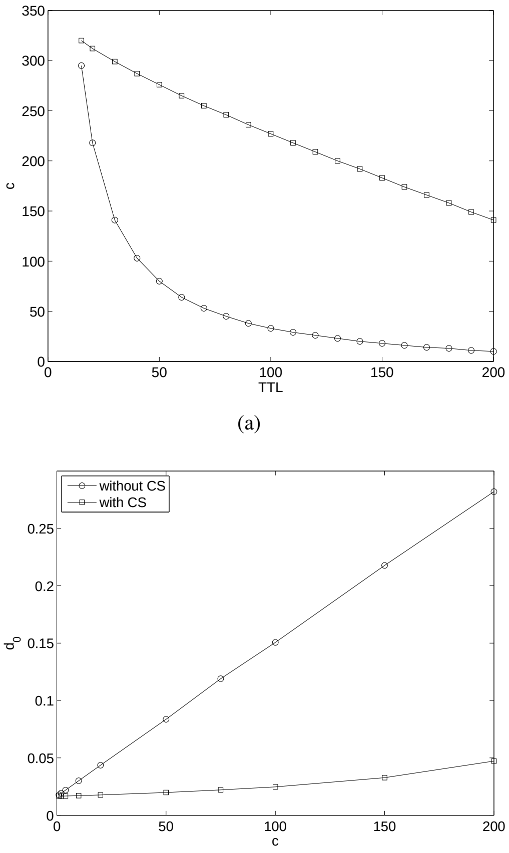
Decentralization can be achieved by letting peers receive information on $x^{(i)}$ from all the others. Information could be disseminated by means of *flooding* or *gossiping*.

In flooding-based dissemination a peer sends its packed non-zero information to its neighbors. This collection of neighbors then forwards the message to their neighbors (excluding, of course, the neighbor that sent the original message). These neighbors may then propagate the message to their neighbors and so on up to a certain predefined maximum level (TTL). To obtain local access to global information each peer starts to flood the overlay network with its own non-zero packed data.

In gossiping peers can store in a buffer a maximum number of messages, a message is forwarded up to a maximum number of times, and each time a peer randomly selects a certain number (called the *fanout*) other peers to forward the message to. Dissemination is achieved by a peer that starts a round of gossiping and in our context each peer starts its own gossiping round. It is proved that atomic reliable broadcasting, i.e., all peers receive the data a peer starts to disseminate, is achieved with high probability if the fanout is on average $O(\log|V|)$ [16] taking $O(\log|V|)$ rounds to complete.

Both schemes have the drawback of introducing a lot of redundancy, i.e., the same message can be received more than once by the same peer, especially for peers with a lot of incoming connections. It means that some or all peers may saturate their available processing and communications capacities; indeed, in the most favorable case, the load factor at each peer is $\frac{|V|\cdot\lambda_r}{\mu}$ which is the same of the centralized solution. For gossiping, this and other issues were already discussed in [17] where the authors make explicit a lot of hidden assumptions that are necessary to ensure robustness of gossip-based protocols and that make gossiping unfit in the context we consider in this paper.

RWs could also be exploited *without* CS. To compare the RW based approach with and without CS we denote as b_m the number of bits to reserve in the message payload a sample of the information. Without CS we consider an optimized coding where a simple prefix code is used to achieve lossless compression of the RW payload. According to such an approach zero values are stored using only one bit prefix code, e.g. 0, and b_m+1 bits are used to store a non-zero value, e.g. 1 followed by the b_m bits of the sample. The identities of the visited edges must be carried by the RW in both cases, requiring b_a bits (if IPv4 addresses of peers are used as identifiers we have $b_a = 32$). It follows that without CS the size of the RW is a random variable since a value associated with an outgoing link is non-zero with probability q; in this case, the average size of the RW is $S_{without} = b_a \cdot TTL + c\cdot(q\cdot b_m+1)\cdot TTL$. The size of the RW with CS after TTL hops is equal to $S_{with} = b_a \cdot TTL + c \cdot (b_m + log_2 TTL)$, where the term $c \cdot (b_m + log_2 TTL)$ represents the cost to accommodate a combination obtained as the sum of TTL values on b_m bits.



(a)

(b)

Fig. 6. Maximum value of c as a function of TTL (a) and d_0 as a function of c with $|E| = 10000$ (b) in the case $q = 0.05$ and $b_m = b_a = 32$.

The first comparison is carried out by considering a maximum size K for the RW, e.g., the typical size of a UDP datagram yields $K = 12000$ bits, a piece of information with sparsity $q = 0.05$ and $b_m = b_a = 32$. In this case the maximum value of c that can be dealt with by the two techniques is obtained by equating S_{with} and $S_{without}$ to K and solving for c. Figure 6(a) shows that for a fixed value of TTL our solution allows for much larger values of c hence it allows to access a larger global information.

Furthermore, we consider the case where the value of c is fixed. We then obtain TTL from equating S_{with} and $S_{without}$ to K and use Equations (2) and (3) with $|E| = 10000$ and $m = \lfloor q \cdot |E| \cdot \log(\frac{1}{q}) \rfloor$ [15] to obtain s_m that guarantees that $\overline{d} > 10$. The pair (s_m, TTL) is used to compute d_0 according to Equation (3) for both techniques. Figure 6(b) clearly shows the superiority of the CS based approach that allows for significantly smaller d_0 values.

Two final remarks are in order: first, each peer only needs to store one sensing matrix Φ for all c information since the elements of Φ in our technique are determined only by the path followed by the RWs. Second, peers constantly receive on average a number of messages that depends only on the number of neighbors and on the parameter w, i.e., it is independent from λ_r and $|V|$. These two characteristics make our approach very scalable.

VIII. Conclusions and future works

In this paper we devised a solution to grant peers local access to global large and sparse information at a given rate. The key ingredients of our technique are CS and RW. The former allows one to collect and compress the information in a distributed fashion; the latter represents a lightweight solution to distribute this compressed information with a controlled communication overhead.

We developed and validated an analytical model to design the parameters of our technique to guarantee high recovery probability. We proved the technique to be feasible by developing and deploying a prototype implementation on PlanetLab.

We are currently working to remove the assumption on the signal sparseness according to the results that show that CS can be used with any *compressible* information.

References

[1] M. Jelasity, A. Montresor, and O. Babaoglu, “Gossip-based aggregation in large dynamic networks,” *ACM Transactions on Computer Systems*, vol. 23, no. 3, pp. 219–252, Aug. 2005.

[2] E. Candès and T. Tao, “Near-optimal signal recovery from random projections: universal encoding strategies?” *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[3] D. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[4] J. Haupt and R. Nowak, “Signal reconstruction from noisy random projections,” *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4036–4048, Sep. 2006.

[5] T. T Bu, J. Cao, A. Chen, and P. Lee, “A fast and compact method for unveiling significant patterns in high speed networks,” in *IEEE INFOCOM 2006*, Apr. 2006.

[6] Y. Lu, A. Montanari, B. Prabhakar, S. Dharmapurikar, and A. Kabbani, “Counter braids: a novel counter architecture for per-flow measurement,” in *ACM SIGMETRICS 2008*, Jun. 2008.

[7] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, “Spatio-temporal compressive sensing and internet traffic matrices,” in *ACM SIGCOMM 2009*, Aug. 2009.

[8] W. Bajwa, J. Haupt, A. Sayeed, and N. R., “Compressive wireless sensing,” in *IPSN*, Apr. 2006.

[9] J. Meng, H. Li, and Z. Han, “Sparse event detection in wireless sensor networks using compressive sensing,” in *43rd Annual Conference on Information Sciences and Systems (CISS)*, 2009, pp. 182–185.

[10] M. Rabbat, J. Haupt, A. Singh, and R. Nowak, “Decentralized compression and predistribution via randomized gossiping,” in *IPSN*, Apr. 2006.

[11] J. Haupt, W. Bajwa, M. Rabbat, and R. Nowak, “Compressed sensing for networked data,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 92–101, Mar. 2008.

[12] R. Berinde, A. Gilbert, P. Indyk, H. Karloff, and S. M.J., “Combining geometry and combinatorics: A unified approach to sparse signal recovery,” *preprint*, 2008.

[13] R. Berinde, P. Indyk, and M. Ruzic, “Practical near-optimal sparse recovery in the l1 norm,” in *Allerton Conference on Communication, Control and Computing*, 2008.

[14] S. Jafarpour, X. W., B. Hassibi, and R. Calderbank, “Efficient and robust compressed sensing using optimized expander graphs,” *IEEE Transactions on Information Theory*, vol. 55, no. 9, pp. 4299 – 4308, Sep. 2009.

[15] J. Tropp and D. Needell, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples,” *Appl. Comput. Harmon. Anal.*, vol. 26, pp. 301–321, 2008.

[16] A. Kermarrec, L. Massoulié, and A. Ganesh, “Probabilistic reliable dissemination in large-scale systems,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 3, pp. 248–258, Mar. 2003.

[17] L. Alvisi and et al, “How robust are gossip-based communication protocols?” *Operating Systems Review*, vol. 41, no. 5, pp. 14–18, Oct. 2007.