A Formal Characterization of Uniform Peer Sampling based on View Shuffling

Yann Busnel, Roberto Beraldi, Roberto Baldoni

A Formal Characterization of Uniform Peer Sampling based on View Shuffling

Yann Busnel, Roberto Beraldi and Roberto Baldoni
Dipartimento di Informatica e Sistemistica
Universitá di Roma – La Sapienza
Via Ariosto 25, 00185 Roma, Italy
Email: name@dis.uniroma1.it

***Abstract*—Consider a group of peers, an ideal random peer sampling service should return a peer, which is an unbiased independent random sample of the group. This paper focuses on peer sampling service based on view shuffling (*aka* gossip-based peer sampling), where each peer is equipped with a local view of size c. This view should correspond to a uniform random sample of size c of the whole system in order to implement correctly a uniform peer sampling service. To this aim, pairs of peers regularly and continuously swap a part of their local views (*shuffling operation*). The paper provides a proof that *(i)* starting from any non-uniform distribution of peers in the peers' local views, after a sequence of pairwise shuffle operations, each local view eventually represents a uniform sample of size c and *(ii)* once previous property holds, any successive sequence of shuffle operations does not modify this uniformity property. This paper also presents some numerical results concerning the speed of convergence to uniform samples of the local views.**

***Keywords*-Peer sampling; Gossip-based protocol; Theoretical analysis; Stochastic process; Numerical evaluation.**

I. Introduction

Uniform peer sampling service has been shown recently to be a basic building block for several applications in large-scale distributed systems [1] as information dissemination [2], counting [3], clock synchronization [4], *etc*. Working on the top of a biased peer sampling can affect either performance, correctness or both of a given application. A sequence of invocations to a peer sampling service returns a sequence of samples of the peers belonging to the system. If samples are unbiased random samples of the system, the peer sampling service is called *uniform*. There are two main approaches to implement uniform random sampling, *random walk* and *gossip-based* protocols.

A random walk on a given graph is a sequential process that consists in visiting the nodes of the graph according to a random order induced by the way the walker is allowed to move. More precisely, the walker moves from one node to one of its neighbors that is selected uniformly at random. The key property of a random walk is that, after a suitable number of steps, called the mixing-time, the visited node is the same as drawn from a uniform distribution [5]. Thus, random walk-based peer sampling mechanisms aim at implementing a biased random walk [3]. Unfortunately, the mixing-time depends on the topological property of the graph, which is generally unknown. Thus, for the reached node to be uniformly sampled, the length of the walk has to be properly tuned. Moreover, this technique may incur in a long delay to return a sample.

This paper focuses on uniform peer sampling based on gossip protocols. We consider a system formed by n peers (*i.e.*, nodes), each provided with a local view of size $c \leq n$. Each node runs a simple *shuffling protocol* where pair of nodes regularly and continuously swaps part of their local views (*shuffle operation*). This protocol is similar to the ones used in [1], [6], [7], [8]. The shuffling protocol aims that local views eventually represent a uniform random sample of the system. The main results presented in this paper show formally that:

1) starting from any non-uniform distribution of nodes in the local views, after a sufficiently long sequence of pairwise shuffle operations executed by the shuffling protocol, each local view represents a uniform random sample of size c among the whole system (Theorem 5.1);
2) once previous property has been established, any sequence of successive shuffle operations does not modify the previous property (Corollary 5.1).

To the best of our knowledge, these results have never been formally proved before, despite the fact that there is empirical evidence shown in many papers [1], [8], that protocols based on view shuffling can provide continuously a uniform sampling.

Let us remark that this result complements the one presented in [9]. Indeed, the authors of [9] propose a protocol based on view shuffling and formally prove that this protocol converges to a uniform peer sampling also in the presence of byzantine peers. Each run of their protocol leads, after a sufficiently long sequence of shuffle operations, to verify the property: "each local view is a uniform random sample of the system". However, each time a user requires to get a new uniform sample, another instance of this protocol has to be started and it has to converge to a new uniform random sample. Conversely, the shuffling protocol presented in this paper shows that once the local view converges to represent a uniform sample of the system, then successive shuffle operations do not modify the property (Corollary 5.1). Therefore, there is a continuous availability of a uniform random sample without the need to start other instances of the base protocol. The paper finally presents some

numerical results related to the shuffling protocol concerning the speed of convergence to uniform samples of the local views.

This paper is organized as follows: Section II presents the system model. The shuffling protocol is presented in Section III while Section IV provides an analytical model of the shuffling protocol. Section V proves that the local views shaped by the shuffling protocol converge to uniform random samples of the system. Section VI provide some stochastic evaluations in order to illustrate the best settings according to the system parameters. Finally, related works and conclusion are given respectively in Section VII and in Section VIII.

II. System model

We consider a finite set of n nodes (with $n \geq 2$), which are uniquely identified through a system-wide identifier (ID). Each node i manages a local partial view of the system, denoted V_i of size $c \leq n$ about all the other nodes in the system, including itself.

The view of node i is modelled as a fixed-size set of binary random variables indicating whenever the identifier k appears in V_i or not

$$X_i = (X_{1i}, X_{2i}, \ldots, X_{ni}) \text{ where } X_{ki} = \begin{cases} 1 & \text{if } k \in V_i; \\ 0 & \text{otherwise} \end{cases}$$

The vector X_i is referred as the *characteristic vector* of the view. A vector X_i is associated with a *probability vector*

$$P_i = (p_{1i}, p_{2i}, \ldots, p_{ni})$$

where p_{ki} is the probability that k belongs to V_i, *i.e.*, $p_{ki} = \mathbb{P}[X_{ki} = 1]$.

The whole system is then modelled as the collection

$$S = (X_1, X_2, \ldots, X_n)$$

of the characteristic vectors corresponding to the nodes' view. The corresponding set of probability vectors is called a *configuration* of the system,

$$\mathcal{C} = (P_1, P_2, \ldots, P_n).$$

Definition 2.1: A view is *uniform* if at a random instant of time all IDs appear in this view with the same probability.

Definition 2.2: A system is called *uniform* if all views are uniform.

In this paper we use the notion of potential function to deal with arbitrary configurations.

Definition 2.3: The *local potential function* of given probability vector P is

$$h(P) = \max_{p_k \in P} \left\{ p_k - \frac{c}{n} \right\}.$$

Definition 2.4: The *potential function* of configuration $\mathcal{C}$ is

$$h(\mathcal{C}) = \max_{P_i \in \mathcal{C}} \{h(P_i)\} = \max_{P_i \in \mathcal{C}} \max_{p_{ki} \in P_i} \left\{ p_{ki} - \frac{c}{n} \right\}.$$

The potential function is a sort distance measure between a generic configuration and the uniform configuration, namely the configuration with all probabilities equal to $\frac{c}{n}$. For such configuration, let us introduce the following lemma. First of all, let consider the following property:

Algorithm 1: Shuffling operation

node i	**node** j
$\ell_i \leftarrow \text{UniRand}(l, V_i)$	$\ell_j \leftarrow \text{UniRand}(l, V_j)$
$V_i' \leftarrow (V_i - \ell_i) \cup \ell_j$	$V_j' \leftarrow (V_j - \ell_j) \cup \ell_i$
$V_i \leftarrow V_i' \cup \text{UniRand}(c - \lvert V_i' \rvert, \ell_i - \ell_j)$	$V_j \leftarrow V_j' \cup \text{UniRand}(c - \lvert V_j' \rvert, \ell_j - \ell_i)$

Property 2.1: The expected size of a view V_i is

$$\mathbb{E}\left[\sum_k X_{ki}\right] = \sum_{k=1}^{n} \mathbb{E}[X_{ki}] = c$$

Lemma 2.1: Let $\mathcal{C}$ be a configuration (*i.e.* distribution of all local views). $h(\mathcal{C})$ is zero $\Leftrightarrow$ $\mathcal{C}$ is uniform.

Proof:

($\Leftarrow$) Let consider the distribution $\mathcal{C}$ as uniform. As all the views are uniform, all the probability for a node to appear in any view is the same, so called $\overline{p}$. Thus, from Property 2.1, we have:

$$c = \sum_{k=1}^{n} \mathbb{E}[X_k] = n \cdot \overline{p} \quad \Longrightarrow \quad \overline{p} = \frac{c}{n}.$$

and then, for all nodes, the local potential is zero. So, $h(\mathcal{C}) = 0$.

($\Rightarrow$) On the other hand, if the potential function is zero ($h(\mathcal{C}) = 0$), then, by definition, the maximum for any probability vector is $\frac{c}{n}$. Thus, from Property 2.1, this also implies that *all* the probabilities are equals to $\overline{p}$, which is the definition of uniformity (*cf.* Definition 2.2). ■

III. The shuffling protocol

We now consider a distributed protocol in which nodes manage their views by performing elementary pairwise shuffle or *shuffling operation*, denoted as $\diamond$. The notation $i \diamond j$ is used to denote that i performs a shuffling operation with j. The effect of an operation is to update the nodes' view, as detailed later in this section. We then show that the protocol makes the system to converge towards a uniform configuration, namely a configuration with zero potential function.

We assume that two shuffles involving a common node may not take place concurrently. Once a node initiates a shuffle, it will be locked until the operation is terminated.

The shuffling operation: The shuffling operation is the core aspect of the whole protocol. The shuffling protocol consists of applying the shuffling operation repeatedly to pairs of nodes i, j (the selection of i and j is explained below).

This shuffling operation has one parameter, the shuffle length l, and involves two views, say V_i and V_j. For the sake of simplicity, we will also use the shuffle ratio, $\gamma = \frac{l}{c}$. The operation $\diamond$ acts as follows.

The view V_i (resp. V_j) is split into two random parts. The first part, denoted as ℓ_i (resp. ℓ_j), is the *sent view*, which is a subset of V_i (resp. V_j) of size l. The elements in ℓ_j are

added to $V_i - \ell_i$, and inversely. If the size of this new set, $V_i' = (V_i - \ell_i) \cup \ell_j$ is lower than c (this could happen if ℓ_j and $V_i - \ell_i$ have common elements), then $l' = c - |V_i'|$ elements are taken from $\ell_i - \ell_j$ at random and added to V_i'. More formally, the shuffling operation consists of the steps presented in Algorithm 1. In the latter, UniRand(h, V) returns a subset of h elements taken uniformly at random from V. The shuffle operation is symmetric in the sense that node j acts exactly as node i. Moreover, the two nodes make their decisions about which elements to keep from the sent view, if any, independently from each other. Thus, the probability of a node k to appear in a view is only determined by the elements in the interacting nodes before the shuffle[1].

Consider the following example. Assume that $c = 7$ and $l = 3$. Consider then a shuffle between the views $V_i = \{0, 12, 1, 5, 3, 7, 8\}$ and $V_j = \{3, 11, 4, 5, 8, 2, 1\}$ with sent subset $\ell_i = \{3, 7, 8\}$, $\ell_j = \{8, 2, 1\}$. We then have that the first manipulation:

$$V_i' = (V_i - \ell_i) \cup \ell_j$$

produces

$$V_i' = \{0, 12, 1, 5\} \cup \{8, 2, 1\} = \{0, 12, 1, 5, 8, 2\}$$

As $|V_i'| = 6$ while $c = 7$, we need to add some random elements of the set

$$\ell_i - \ell_j = \{3, 7, 8\} - \{8, 2, 1\} = \{3, 7\}$$

A remark on system partitioning: For the sake of simplicity and without loss of generality to carry out our analysis, we assume that any shuffle operation does not partition the system. The shuffling protocol presented above could indeed lead temporary to system partitioning with a small probability. Some practical solution to this problem has been proposed in [8]. These solutions lead to asymmetric shuffling operations that do not affect the results of Section V, as proved in [10].

IV. Protocol analysis

In this section, we derive an analytical model of the shuffling protocol, which captures the variation of the system configuration over time. The main symbols used in this paper are reported in Table I.

A. View evolution

Let consider how the presence of element k in the view of i varies after a shuffling operation among nodes i and j. A shuffling operation between i and j, denoted $i \diamond j$, generates two new characteristic vectors, X_i' and X_j', starting from the original vectors X_i and X_j. In other words, after the operation, the view of node i (resp. j) is described by X_i' (resp. X_j').

The evolution of the view over time is then described by a relationship among X and X'. Before describing this relationship, it is important to understand that X_{ki}' is independent from the others random variables X_{kj}'. In fact, the elements that are inserted or removed due to shuffling into the view of node i, are not influenced by elements inserted/removed into the view of node j. In other words, as explained above, i and j do not coordinate somehow about their decisions on the way to change the views. Node i and j act locally and then, independently from each other. Let P_{10} be the probability that, after the shuffle, node k is removed from V_i and P_{01} the probability that k is inserted (for the sake of simplicity indexes are omitted), namely

$$P_{10} = \mathbb{P}[X_{ki}' = 0 | X_{ki} = 1] \quad \text{and} \quad P_{01} = \mathbb{P}[X_{ki}' = 1 | X_{ki} = 0]$$

The probability that k appears in V_i, given that $i \diamond j$, is then

$$\mathbb{P}[X_{ki}' = 1 | i \diamond j] = (1 - \mathbb{P}[X_{ki} = 1])P_{01} + \mathbb{P}[X_{ki} = 1](1 - P_{10}) \tag{1}$$

This expression has the following meaning. The probability that node k appears in i's view, after a shuffle between i and j, is given by the probability that k was not in the view and it has been added or the probability that k was already in the view and it has not been deleted. The evolution of a view is best described as a two states Markov chain, see Figure 1, where state 1 (resp. state 0) means that k is (resp. not) in the node i's view.

Node i receives l elements from j. As an element is sent with probability γ, the expected number of elements that ℓ_j and V_i have in common is:

$$\sum_k \mathbb{P}[X_{ki} = 1] \cdot \gamma \cdot \mathbb{P}[X_{kj} = 1]$$

n	Total number of nodes in the system
c	Size of local view
l	Size of the sent vector
γ	Shuffle ratio ($\gamma = \frac{l}{c}$)
V_i	Local view of node i
ℓ_i	Sent view of node i
X_{ki}	Indication function
X_i	Characteristic vector of view V_i
P_i	Probability vector of view V_i
$\overline{p}$	Expected uniform probability ($\overline{p} = \frac{c}{n}$)
M_{ij}	Expected number of shared elements between i and j
$i \diamond j$	Suffling operation (i shuffles with j)

TABLE I
List of main symbols

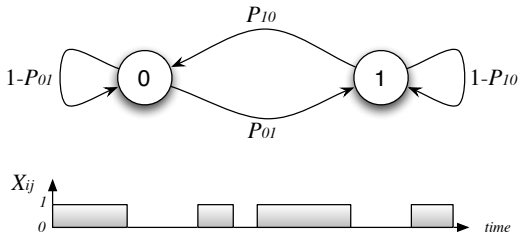

Fig. 1. Markov chain representing the evolution of X_{ij}, the presence of element i into the view of node j. When the view is uniform, the fraction of time that the element appears in the view, (*i.e.*, $X_{ij} = 1$) is the same for any element. This condition corresponds to the steady state of the chain.

[1]A correlation would arise if, for example, node i decides to add the identifier k received from j only if j promises something else back.

$$= \gamma \cdot \sum_k \mathbb{P}[X_{ki} = 1]\mathbb{P}[X_{kj} = 1] = \gamma \cdot M_{ij}$$

Now, the view size must remain constant. The expected number of elements removed from i must then be equal to the number of new elements added into V_i, which is equal to $l - \gamma M_{ij}$. As all elements have to equally likely be removed, the probability to remove the element k is $\frac{l-\gamma M_{ij}}{c}$. From which:

$$P_{10} = \gamma \cdot \left(1 - \frac{\sum_k \mathbb{P}[X_{kj} = 1]\mathbb{P}[X_{ki} = 1]}{c}\right)$$

On the other hand, as element k can be added only if it belongs to ℓ_j, we have:

$$P_{01} = \gamma \cdot \mathbb{P}[X_{kj} = 1].$$

B. Evolution of the system

Let now consider how the system evolves. As explained above, we assume that concurrent operations cannot occur. Thus, we can serialize parallel shuffles in an arbitrary order and assume that only one shuffling operation may take place at a time. Let $P_{ex}(i, j)$ be the probability that i and j make the shuffle, *i.e.*, $P_{ex}(i, j)$ is the probability that the operation $i \diamond j$ takes place.

We can describe the global evolution of the system with the following expression:

$$\mathbb{P}[X'_{ki} = 1] = \sum_j P_{ex}(i, j) \cdot \mathbb{P}[X'_{ki} = 1 | i \diamond j] \tag{2a}$$

$$+ \sum_j P_{ex}(j, i) \cdot \mathbb{P}[X'_{ki} = 1 | j \diamond i] \tag{2b}$$

$$+ \left(1 - \sum_j (P_{ex}(i, j) + P_{ex}(j, i))\right) \cdot \mathbb{P}[X_{ki} = 1] \tag{2c}$$

This last equation means that the probability vector of a node follows the view evolution presented in Equation 1 if it is involved in a view shuffle (Equation 2a and 2b) and remains the same if it is not involved in the last shuffle (Equation 2c).

V. Convergence property of the protocol

Let now show that the shuffling protocol makes the system to converge towards a uniform configuration. In particular, we show that if the shuffling protocol is executed by a system with arbitrary view distribution, then eventually the system converges towards a uniform configuration, *i.e.*, a system in which all the local views represent uniform random samples of the system. In order to show this result, we exploit the notion of potential function, introduced in Section II. We will show that if the potential function of a configuration is greater than zero, then after a shuffling operation the potential function of the configuration is reduced. Roughly speaking, this means that a shuffling operation moves the system towards a "more" uniform system, or makes the system closer to the uniform configuration. Formally, we have:

Lemma 5.1 (Operator $\diamond$ reduces the potential): Let P and Q be two probability vectors of nodes i and j and let P', Q' these vectors after a shuffling operation $i \diamond j$. Then $\max\{h(P'), h(Q')\} < \max\{h(P), h(Q)\}$.

Proof: For the sake of simplicity, we denote the maximum probability before the shuffle as $p^* = \max\{h(P), h(Q)\}$. We prove below that $\forall k \in [1..n]$, (1) $\Delta p_k = p'_k - p^* < 0$, and that (2) $\Delta q_k = q'_k - p^* < 0$ where p_k, q_k, p'_k and q'_k denoted respectively $P[k], Q[k], P'[k]$ and $Q'[k]$. This means that the highest probability decreases and no other probabilities can become greater than the previous maximum.

(1) We want to prove that $\Delta p_k < 0$. From equation 1, we have:

$$p'_k = (1 - p_k) \cdot \gamma \cdot q_k + p_k \cdot \left(1 - \gamma + \gamma \frac{M}{c}\right)$$

$$\implies \Delta p_k = p'_k - p^* = p_k + \gamma \left(q_k - q_k \cdot p_k - p_k + p_k \cdot \frac{M}{c}\right) - p^*$$

As $M = \sum_k p_k \cdot q_k < \sum_k p^* \cdot q_k = p^* \cdot c$ and $\gamma \leq 1$, we have:

$$\begin{aligned}
\Delta p_k &\leq q_k - q_k \cdot p_k + p_k \cdot \frac{M}{c} - p^* \\
&< q_k - q_k \cdot p_k + p_k \cdot p^* - p^* \\
&= q_k(1 - p_k) - p^*(1 - p_k) = (q_k - p^*) \cdot (1 - p_k) \\
&\leq 0 \qquad \text{as } \forall i, \quad p_k \leq p^* \leq 1 \text{ and } q_k \leq p^*.
\end{aligned}$$

(2) It remains to prove the same upper bound for Q', *i.e.* $\forall i, \Delta q_k = q'_k - p^* < 0$. According to the Equation 1, by symmetry, we have:

$$\mathbb{P}[X'_{ki} = 1 | j \diamond i] = q_k + \gamma \left(p_k - p_k \cdot q_k - q_k + q_k \cdot \frac{M}{c}\right).$$

Thus, following the same reasoning, we obtain that $\forall i, \Delta q_k < 0$.

Therefore, we can conclude that $\max\{h(P'), h(Q')\} < \max\{h(P), h(Q)\}$ and the claim follows. ∎

We are now in the position to state the following theorem:

Theorem 5.1 (Convergence to uniformity):
Let i be the number of shuffling operations executed on a system of n nodes, $\mathcal{C}_0$ be any initial unpartitioned distribution of local views and $\mathcal{C}_i$ be the configuration of the system after those i shuffling operations. Local views built by the shuffling protocol presented in Section III will converge to uniform random samples of the system, *i.e.*,

$$\forall \mathcal{C}_0, \lim_{i \to \infty} h(\mathcal{C}_i) = 0;$$

Proof: The claim follows from result comes from Lemma 5.1, as a shuffling operation *strictly* reduce the local potential of the pair involved in the shuffle. Thus, the distance of the current distribution of sample with the uniformity could only monotonically reduce, due to Equation 2. Then, the distribution of the samples converges to the uniform one due to Lemma 2.1. ∎

Let us now show a corollary stating that once local views represent uniform samples of the system, the shuffling protocol keeps this property true forever.

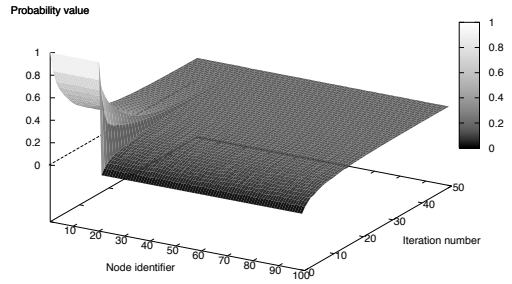

Fig. 2. Evolution of $\mathbb{P}[X_{ji} = 1]$ for i fixed according to the gossip cycle iteration. – *Settings:* $n = 100$*,* $c = 20$ *and* $l = 4$ *for 50 iterations.*

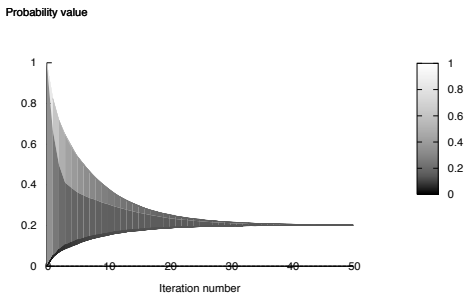

Fig. 3. Evolution of $\mathbb{P}[X_{ji} = 1]$ for i fixed according to the gossip cycle iteration (Planar view). – *Settings:* $n = 100$*,* $c = 20$ *and* $l = 4$ *for 50 iterations.*

Corollary 5.1 (Operator $\diamond$ preserves uniformity): Let $\mathcal{C}$ be a uniform unpartitioned distribution of local views. A shuffling operation executed by the shuffling protocol presented in Section III between any pair of two local views X_i and X_j belonging to $\mathcal{C}$ produces a distribution $\mathcal{C}'$ that is uniform.

Proof: Lemma 5.1 gives us that the potential of two views involved in a shuffling operation can only decrease. Given the fact that $\mathcal{C}$ corresponds to the uniform distribution, X_i and X_j are uniform and $P_i = P_j$ are vectors with all elements equal to $\overline{p} = \frac{c}{n}$. Thus, due to Lemma 2.1 and Definition 2.4, the potential of X_i and X_j are $h(P_i) = h(P_j) = 0$. From Lemma 5.1, after the shuffle, $h(P_i)$ and $h(P_j)$ cannot increase and thus, remain to 0. Then, $\mathcal{C}'$ is the uniform distribution. ■

VI. Numerical results

In this section, we apply the analytical model (*cf.* Equation 2) in order to numerically derive some representative evolutions of a system, in which shuffles are organized into cycles. One cycle corresponds to all nodes initiate exactly one shuffle with a random partner chosen from its own view[2].

Consider a system with view size $c = 20$. Initially, the views of nodes are set to $[1..20]$. This corresponds to one of the worst cases of starting state. Indeed, among a population of 100 nodes, the identifiers $[21..100]$ do not appeared in any view at starting point. They will be introduced progressively by the initiator of the shuffle. For example, when node 21 initiates an exchange with node 1, node 1 becomes aware of 21 if 21 sends its ID (to avoid partitioning, we simulated the mechanism described in [8], *i.e.*, instead of sending its own ID with probability $\frac{l}{c}$ the initiator of the shuffle sends its ID with probability 1).

Figure 2 shows the view evolution of one node. The z-axis shows the probability that an ID appears in the view of this node.

At the beginning of an execution, the overlay is then fixed: c nodes have a probability equals to 1 to appear in a view and the other nodes a probability equals to 0. When the protocol runs, all nodes are proceeding to theirs shuffles during each cycle. Figures 2 shows the evolution of each probability $\mathbb{P}[X_{ji} = 1]$ according to the node identifier j and the iteration of the algorithm, where one iteration corresponds to one gossip cycle.

[2]This cycle-based behavior is well-known in gossip-based protocols [1], [6], [7], [8].

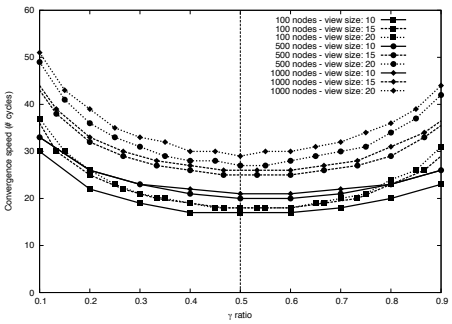

Fig. 4. Number of gossip cycles required to reach uniformity for different settings, according to the ratio γ.

Figures 3 represents the same data but in a planar view (all the probabilities of each time are mapped vertically). Thus, the latter figure shows the evolution of the maximum (and *a fortiori* the minimum) probability value at a given time. It is possible to observe that all the probabilities converge to the average value ($\frac{c}{n} = 0.2$) in less than 40 gossip cycles.

In order to evaluate the impact of the system parameters, Figure 4 presents the average convergence time required to reach the uniform sampling, according to the ratio $\gamma = \frac{l}{c}$, for different settings of the system (from 100 to 1,000 nodes with a view size varying from 10 to 20), starting from the same aforementioned worst case. This figure speaks about how to obtain the best convergence time according to γ. Independently of the size of the network, the size of the view c and the initial state, the fastest convergence is obtained with a ratio $\gamma = 0.5$ (represented by a vertical line on Figure 4). Thus, in the design of a gossip-based protocol, l has to be set to the half of c in order to obtain the highest efficiency in term of convergence speed.

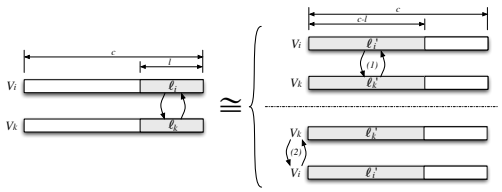

Fig. 5. Intuitive equivalence between a small ℓ value and the opposite $c - \ell$ ones.

This conjecture can be proved as sketched below. A shuffle operation with a sent vector ℓ between two nodes is equivalent to a shuffle with the complementary of ℓ (*i.e.* $V - \ell$), followed by swapping the ID of these two nodes (*cf.* Figure 5). Indeed, in this figure, the content of V_i after the shuffle on the left side is equivalent to the content of V_i on the right side after *(1)* a shuffle with the sent vector $\ell'_i = V_i - \ell_i$ and *(2)* swapping the node's ID (i becomes k and *vice versa*).

Now, consider $l \leq \frac{c}{2}$. It is obvious that the higher the size of the sent vector, the greater the effectiveness[3] of a shuffle. Moreover, according to the above equivalence, a shuffle with l is equivalent to a shuffle with $c-l$. Thus, for $l \geq \frac{c}{2}$, the lesser the size of the sent vector, the greater the effectiveness of a shuffle. So, the greatest effectiveness is reached for $l = \left\lfloor \frac{c}{2} \right\rfloor$, as confirmed numerically in Figure 4.

VII. Related works

Apart of the paper presented in [9] that we discussed in Section I and that remains the one that presents the result closest to ours, several contributions have been proposed in the context of gossip-based peer sampling service [1], [8]. In these works, authors propose and study the same framework than the ones we modelled in this paper (as known as gossip-based protocol). Although, the evaluation of their samples' distribution is conducted only using empirical experimentations. To the best of our knowledge, no fully theoretical analysis of the shuffling protocol with respect to sampling uniformity has been proposed so far.

Several contributions provided some fully theoretical analysis of gossip-based protocols as [1], [11], [12], [13]. However, those analysis aims to provide some theoretical outcomes on a specific characteristic of these protocols as convergence speed of dissemination protocols, by defining precise lower and upper bound of the mixing time, degree balancing, *etc.* Nevertheless, in these works, authors do not consider the local view as the information to analyse. In their works, the network is modelled as a probabilistic matrix, which represents the meeting probability of any pair of peers, and this matrix is used as a building block of their analyses. Our study can then be used to provide this specific matrix and/or to confirm that the matrix used in these related works are consistent with the real behavior of gossip-based protocols.

As remarked in Section I, random walks have been also used to provide uniform peer sampling [3], [14]. These contributions proposed how to bias the simple random walks model in the way to extract uniform sampling. Both of them provide a theoretical analysis of their protocols. Finally, a solution of the peer sampling service, based on a structured P2P system, has been proposed in [15]. Authors propose an algorithm based on Chord [16] and proved that it provides nodes with uniform random samples of the system.

[3]Roughly speaking, *effectiveness* represents how different the shuffled views are from the ones before the shuffle. The higher the difference, the greater the effectiveness.

VIII. Concluding Remarks

The paper has provided a theoretical ground to the fact that a shuffling protocol provides eventually nodes with uniform random samples of a system. Before this was only an empirical evidence. Differently from [9], our analysis shows that the same instance of the shuffling protocol can provide permanently a node with uniform sample of the system. Corollary 1 formally grasps this difference. The paper also presented a numerical evaluation of the shuffling algorithm on its convergence speed of the local views to uniform random samples and also what is the best fraction of the local views to swap in a shuffling operation to get best convergence speed.

Acknowledgment

We would like to warmly thank Leonardo Querzoni for his help with the simulations.

References

[1] M. Jelasity, S. Voulgaris, R. Guerraoui, A.-M. Kermarrec, and M. van Steen, "Gossip-based peer sampling," *ACM Transaction on Computer System*, vol. 25, no. 3, p. 8, august 2007.

[2] R. Baldoni, R. Beraldi, V. Quema, L. Querzoni, and S. Tucci-Piergiovanni, "TERA: topic-based event routing for peer-to-peer architectures," in *1st int'l conf. on Distributed Event-Based Systems (DEBS '07)*. Toronto, Ontario, Canada: ACM, 2007, pp. 2–13.

[3] L. Massoulié, E. L. Merrer, A.-M. Kermarrec, and A. Ganesh, "Peer counting and sampling in overlay networks: Random Walk Methods," in *the 25th annual ACM symposium on Principles of distributed computing (PODC '06)*, Denver, CO, USA, july 2006, pp. 123–132.

[4] R. Baldoni, A. Corsaro, L. Querzoni, S. Scipioni, and S. T. Piergiovanni, "Coupling-Based Internal Clock Synchronization for Large Scale Dynamic Distributed Systems," *IEEE Transactions on Parallel and Distributed Systems*, To appear.

[5] B. Bollobás, *Random Graphs – 2nd Ed.* Cambridge Univ. Press, 2001.

[6] P. T. Eugster, S. Handurukande, R. Guerraoui, A.-M. Kermarrec, and P. Kouznetsov, "Lightweight probabilistic broadcast," *ACM Transactions on Computer Systems*, vol. 21, no. 4, pp. 341–374, novembre 2003.

[7] M. Jelasity and O. Babaoglu, "T-Man: Fast Gossip-based Construction of Large-Scale Overlay Topologies," University of Bologna, Dpt. of CS, Bologna, Italy, Research Report UBLCS-2004-7, may 2004.

[8] S. Voulgaris, D. Gavidia, and M. van Steen, "CYCLON: Inexpensive Membership Management for Unstructured P2P Overlays," *Journal of Network System Management*, vol. 13, no. 2, pp. 197–217, june 2005.

[9] E. Bortnikov, M. Gurevich, I. Keidar, G. Kliot, and A. Shraer, "Brahms: byzantine resilient random membership sampling," in *the 27th ACM symposium on Principles of Distributed Computing (PODC '08)*. Toronto, Canada: ACM, 2008, pp. 145–154.

[10] Y. Busnel, R. Beraldi, and R. Baldoni, "A Formal Characterization of Uniform Peer Sampling based on View Shuffling," MIDLAB, Tech. Rep. 4/09, June 2009.

[11] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking, "Randomized rumor spreading," in *the 41st Annual Symposium on Foundations of Computer Science (FOCS '00)*. IEEE Computer Society, 2000, p. 565.

[12] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-Based Computation of Aggregate Information," in *the 44th IEEE Symposium on Foundations of Computer Science (FOCS '03)*, octobre 2003, pp. 482–491.

[13] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. on Networks*, vol. 14, pp. 2508–2530, 2006.

[14] M. Zhong, K. Shen, and J. Seiferas, "Non-uniform random membership management in peer-to-peer networks," in *24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 2, March 2005, pp. 1151–1161 vol. 2.

[15] V. King and J. Saia, "Choosing a random peer," in *the 23rd annual ACM symposium on Principles of Distributed Computing (PODC '04)*. New York, NY, USA: ACM, 2004, pp. 125–130.

[16] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Transaction on Networks*, vol. 11, no. 1, pp. 17–32, 2003.