On the Power of the Adversary to Solve the Node Sampling Problem

Emmanuelle Anceaume, Yann Busnel, Sébastien Gambs

On the Power of the Adversary to Solve the Node Sampling Problem

Emmanuelle Anceaume[1], Yann Busnel[2], and Sébastien Gambs[3]

[1] IRISA / CNRS Rennes (France), emmanuelle.anceaume@irisa.fr
[2] LINA / Univ. de Nantes (France), Yann.Busnel@univ-nantes.fr
[3] IRISA / Univ. de Rennes 1 - INRIA Rennes – Bretagne Atlantique (France), sebastien.gambs@irisa.fr

Abstract. We study the problem of achieving uniform and fresh peer sampling in large scale dynamic systems under adversarial behaviors. Briefly, uniform and fresh peer sampling guarantees that any node in the system is equally likely to appear as a sample at any non malicious node in the system and that infinitely often any node has a non-null probability to appear as a sample of honest nodes. This sample is built locally out of a stream of node identifiers received at each node. An important issue that seriously hampers the feasibility of node sampling in open and large scale systems is the unavoidable presence of malicious nodes. The objective of malicious nodes mainly consists in continuously and largely biasing the input data stream out of which samples are obtained, to prevent (honest) nodes from being selected as samples. First, we demonstrate that restricting the number of requests that malicious nodes can issue and providing a full knowledge of the composition of the system is a necessary and sufficient condition to guarantee uniform and fresh sampling. We also define and study two types of adversary models: (1) an omniscient adversary that has the capacity to eavesdrop on all the messages that are exchanged within the system, and (2) a blind adversary that can only observe messages that have been sent or received by nodes it controls. The former model allows us to derive lower bounds on the impact that the adversary has on the sampling functionality while the latter one corresponds to a more realistic setting. Given any sampling strategy, we quantify the minimum effort exerted by both types of adversary on any input stream to prevent this sampling strategy from outputting a uniform and fresh sample.

Keywords: Data Stream; Kullback-Leibler Divergence; Uniform sampling; Freshness; Byzantine adversary

1 Introduction

We tackle the problem of achieving node sampling in large scale open systems in presence of adversarial (Byzantine) nodes. Uniform sampling is a fundamental primitive guaranteeing that any node in a population has the same probability

to be selected as sample. This property is of utmost importance in systems in which both the population is continuously evolving and it is impossible to capture the full complexity of the network through global snapshots. By collecting random subsets of information over the network, one can infer at almost no cost global characteristics of the whole population such as its size, its topological organization or its resources. Providing at any time randomly chosen identifiers of nodes in the system is an essential building block to construct large scale distributed applications. Uniform sampling finds its root in many problems such as data collection, dissemination, load balancing, and data-caching. A typical example is load balancing in cluster-based applications, in which choosing a host at random among the available ones is often a choice that provides performance close to that offered by more complex selection criteria [1]. Another example is epidemic-based applications, in which periodically selecting a few random nodes as neighbors, preserve the connectivity in large-scale environments despite the dynamical aspect of these systems [2,3,4,5,6].

Providing unbiased (*i.e.*, uniform) sampling in these open systems is a challenging issue. First, this primitive must cope with the continuous change of the network structure caused by nodes departures and arrivals. Furthermore, an important issue that seriously hampers the feasibility of uniform sampling in open and large scale systems is the unavoidable presence of malicious nodes. Malicious (also called Byzantine) nodes typically try to manipulate the system by exhibiting undesirable behaviors [7]. In our context, they try to subvert the system by launching targeting attacks against nodes in the aim of biasing uniformity by isolating honest nodes within the system. This can be quickly achieved by poisoning local views of honest nodes with malicious node ids (*cf.* Section 3.2). For instance in unstructured graphs, a number of push operations logarithmic in the size of local views is sufficient to fully eclipse honest nodes from the local view of a node [8], while in structured graphs, a linear number of join operations is required [9].

Recent works have been proposed to detect and exclude these adversarial behaviors [10,11] by observing that malicious nodes try to get an in-degree much higher than honest nodes in order to isolate them. Extensive simulations [10] have shown that this approach is only highly effective for a very small number of malicious nodes (*i.e.*, in $\mathcal{O}(\log |\mathcal{S}|)$ where $|\mathcal{S}|$ is the size of the network $\mathcal{S}$). Otherwise detection mechanisms can display a high rate of false positives (*i.e.*, detect honest nodes as faulty ones).

When the system is harmed by a large number of malicious nodes (*e.g*, a linear proportion of the nodes of the system), which is definitively a realistic assumption in peer-to-peer systems [7,12], additional mechanisms are required to prevent targeted attacks from succeeding. Specifically, in structured peer-to-peer systems, analytical studies have shown that artificially inducing churn allows to defend the system against adversarial behaviors, either through competitive induced churn strategies [13], or through global induced churn [14]. Briefly, the "induced churn" principle states that, by forcing nodes to periodically change their position in the graph, malicious nodes cannot predict the evolution of the

state of the system after a given sequence of join and leave operations. By taking advantage of the properties of structured graphs, the authors of both papers have shown that, with high probability, any node is equally likely to appear in the local view of each other honest node after a number of rounds polynomial in the size of the system.

Unfortunately, in unstructured peer-to-peer systems, nodes cannot rely on the topological nature of structured graphs to reject new node ids that do not conform to the imposed distance function in contrast to structured networks [13,15]. To circumvent this issue, Bortnikov *et al.* [8] rely on the properties of min-wise independent permutations to eventually converge towards uniform sampling on the node ids. However by construction, this convergence is definitive in the sense that once a random sample has been locally observed it is kept forever as the local sample, thus making this solution static. Intuitively, this lack of adaptivity seems to be the only defense against adversarial behavior when considering bounded resources (*i.e.*, memory and bandwidth).

In this paper we propose a solution that guarantees that each node id received infinitely often has a non-null probability to locally appear as a sample. We coin this property as the *freshness sampling* property, which we formally define later in the paper. We present a formal analysis of the conditions under which uniform and fresh sampling is feasible. More precisely, the first contribution of this paper is to show necessary and sufficient conditions under which uniform and fresh sampling is achievable in unstructured peer-to-peer systems potentially populated with a large proportion of Byzantine nodes. Let $\mathcal{S}$ represent the collection of nodes in the system, and $k < 1$ the proportion of malicious nodes in $\mathcal{S}$. Let δ be the number of (not necessarily unique) malicious node ids gossiped by malicious nodes during a time interval T_s, and Γ_{v_i} denotes the local memory of any honest node v_i in $\mathcal{S}$. We prove the following assertions.

- If the number δ of (non-unique) malicious ids received at node v_i during a given period of time T_s is strictly greater than $T_s - |\mathcal{S}|(1-k)$ then, neither uniform sampling nor fresh sampling can be achieved.
- If $\delta \leq T_s - |\mathcal{S}|(1-k)$ and the size of the memory Γ_{v_i} is greater than or equal to $|\mathcal{S}|$ then, both uniform and fresh sampling can be achieved.
- If $\delta \leq T_s - |\mathcal{S}|(1-k)$, and $|\Gamma_{v_i}| < |\mathcal{S}|$ then, both uniform and fresh sampling cannot be achieved.

Briefly, these conditions show that if the system cannot limit the number of messages an adversary can periodically send, then solving either uniform sampling or fresh sampling is impossible. On the other hand, if this assumption holds and if all honest nodes in the system have access to a very large memory (*i.e.*, linear in the size of the network), then the sampling problem becomes trivially solvable. Unfortunately, we show that both conditions are necessary and sufficient to solve the uniform and fresh sampling problem. As a consequence, these strong conditions highlight the possible damages that adversarial behaviors can cause in large-scale unstructured systems.

We also propose a characterization of the adversarial power towards biasing uniform and fresh sampling. By adopting a statistical view of the input stream

and by comparing distributions using metrics such as information divergence, we derive lower bounds on the work that the adversary has to exert to bias this input stream so that uniform and fresh sampling do not hold. We define and study two models of adversary: (1) the omniscient adversary, which has the capacity to eavesdrop on all the messages that are exchanged within the system, and (2) the blind adversary, which can only observe messages that have been sent or received by malicious nodes. To the best of our knowledge, we are not aware of any previous work that has characterized the minimum effort an adversary has to exert to prevent the uniform and fresh sampling to be achievable.

The outline of this paper is the following. First in Section 2, we give an overview of the existing related work on uniform sampling before describing in Section 3, the system model and the assumptions that we make. In Section 4, we present the functionalities of a sampling component and the properties that it should guarantee. Afterwards in Section 5, we identify the two conditions under which uniform and fresh sampling is achievable, while in Section 6 we review some background on information divergence of data streams. The omniscient and blind adversary models, as well as the characterization of the minimum effort the adversary has to exert to bias the sampling properties, are respectively studied in Sections 7 and 8. Finally, we conclude in Section 9 with some open issues.

2 Related Work

In absence of malicious behaviors, uniform sampling can be achieved through gossip-based algorithms [16,17,18] or through random walks [3,5,6,19]. Gossip-based algorithms mainly consist, for each node v_i in the system, in periodically selecting some other node v_j in v_i's local view and exchanging information. Information can either be pushed to or pulled from other nodes. Over time, information spreads over the system in an epidemic fashion allowing each node to continuously update its local view with fresh node ids. On the other hand, a random walk on a network is a sequential process, starting from an initial node v_i, which consists in visiting a node in v_i's neighborhood according to some randomized order. In its simpler form, the next visited node is chosen uniformly at random among the neighbors, while more sophisticated choices are implemented to cope with the bias introduced by topology, specifically towards high degree nodes (for instance, through the Metropolis-Hastings algorithm [20]). In the literature, different approaches have been proposed to deal with malicious behaviors, each one focusing on a particular adversarial strategy. In this section, we provide a brief overview of these techniques.

In presence of malicious behaviors, different approaches have been proposed according to the considered attacks. Specifically, with respect to eclipse attacks a very common technique, called *constrained routing table*, relies on the uniqueness and impossibility of forging nodes identifiers. This technique selects as neighbors only the nodes whose identifiers are close to some particular points in the identifier space [21]. Such an approach has been successfully implemented into several overlays (*e.g.*, CAN [22], Chord [23] or Pastry [24]). More generally, to pre-

vent messages from being misrouted or dropped, the seminal works of Castro *et al.* [21] and Sit and Moris [7] on Distributed Hash Tables (DHT)-based overlays combine routing failure tests and redundant routing to ensure robust routing. Their approach has then been successfully implemented in different structured-based overlays (*e.g.*, [25,26,27]). In all these previous works, it is assumed that at any time and anywhere in the overlay, the proportion of malicious nodes is bounded and known, allowing powerful building blocks such as Byzantine tolerant agreement protocols to be used among subsets of nodes [26,27].

When such an assumption fails, additional mechanisms are needed. For instance, Scheideler *et al.* [13] propose the *Cuckoo&flip* strategy, which consists in introducing local induced churn (*i.e.*, forcing a subset of nodes to leave the overlay) upon each join and leave operations. This strategy prevents malicious nodes from predicting the exact shape of the overlay after a given sequence of join and leave operations. Subsequently to this theoretical work, experiments have been conducted to verify the practical feasibility of global induced churn, which consists in having all the nodes of the overlay periodically leaving their positions [28]. For instance, researchers [15] have analyzed several adversarial strategies, and have shown that an adversary can very quickly subvert DHT-based overlays by simply never triggering leave operations.

Jesi *et al.* [10] propose a random sampling algorithm taking explicitly into account malicious nodes. Their solution assumes that the ultimate goal of the malicious nodes is to mutate the random graph into a hub-based graph, hub for which malicious nodes gain the lead. Once this goal is reached, malicious nodes can very quickly and easily subvert the whole overlay by performing denial-of-service attacks. Conducting a hub attack mainly consists for malicious nodes in increasing their in-degree. Jesi *et al.* [10] propose to detect highly popular nodes by extending classic peer sampling services with a module that identifies and blacklists nodes that have an in-degree much higher than the other peers of the overlay. This approach, also adopted in several structured based systems [11] through auditing mechanisms, or in sensor networks [29], is effective only if the number of malicious nodes is very small with respect to the size of the system (*i.e.*, typically of $O(\log |\mathcal{S}|)$).

Bortnikov *et al.* [8] have recently proposed a uniform but non-fresh node sampling algorithm tolerating up to a linear number of malicious nodes. This sampling mechanism exploits the properties offered by min-wise permutations. Specifically, the sampling component is fed with the stream of node identifiers periodically gossiped by nodes, and outputs the node identifier whose image value under the randomly chosen permutation is the smallest value ever encountered. Thus eventually, by the property of min-wise permutation, the sampler converges towards a random sample. By limiting the number of requests malicious nodes can periodically issue, their solution requires a single node id to be stored in the local memory. Nevertheless, this approach does not satisfy the freshness property as convergence toward a random sample is definitive.

In previous papers whose results form the basis of this article, Anceaume, Busnel and Gambs [30,31] have shown that imposing strict restrictions on the

number of messages sent by malicious nodes during a given period of time and providing each honest node with a very large memory (*i.e.*, proportional to the size of the system) are necessary and sufficient conditions to obtain uniform and fresh (*i.e.*, non definitive) sampling. These findings complement two previous results [32,33], in which an analysis of the class of uniform and fresh sampling protocols is presented. Both previous works provide a complete analytical proof of a gossip-based protocol achieving both uniformity and freshness. However, in contrast to the present work, adversarial behaviors were not considered.

Finally, taking a completely different approach from the previously mentioned papers based on gossip algorithms or on properties on distance functions, the techniques presented in [34,35] rely on social network topologies to guard against Sybil attacks. Both protocols take advantage of the fact that Sybil attacks try to alter the fast mixing property of social networks to defend against these attacks. However, in presence of malicious nodes with a high degree, the performance of both protocols degrade drastically. Note that the analysis presented in this paper is independent from the way the stream of node ids at each node v_i has been generated. For instance, it may result from the propagation of node ids through gossip-based algorithms (namely through push, pull or push-pull mechanisms initiated by v_i and its neighbors), from the node ids received during random walks initiated at v_i, or even from the induced churn imposed in structured-based overlays.

3 System Model

3.1 Model of the Network

We consider a dynamic system $\mathcal{S}$ populated by a large collection of nodes in which each node is assigned a unique and permanent random identifier from an l-bits identifier space. Node identifiers (simply denoted *ids* in the following) are derived by applying some standard strong cryptographic hash functions on some intrinsic characteristics of nodes. The value of l (*e.g.*, 160 for the standard SHA-1 hash function) is chosen to be large enough to make the probability of identifiers collision negligible. The system is subject to churn, which is classically defined as the rate of nodes' turnover in the system [36]. Each node knows only a small set of nodes existing within the system and this knowledge generally varies according to the activity of the system. The particular algorithm used by nodes to update this small set and to route messages induces the resulting overlay topology. In this work, we consider only unstructured overlays, which are assumed to conform with random graphs, in the sense that relationships among nodes are mostly set according to a random process.

3.2 Model of the Adversary and Security Mechanisms

A fundamental issue faced by any practical open system is the inevitable presence of nodes that try to manipulate the system by exhibiting undesirable behaviors [7,21]. Such nodes are called malicious or Byzantine nodes. In contrast, a

node that always follows the prescribed protocols is called *honest*. Honest nodes cannot *a priori* distinguish honest nodes from malicious ones, which would otherwise render the problem trivial. In our context, manipulating the system amounts to dropping messages that should normally be relayed by malicious nodes towards honest ones and injecting new messages. Injecting new messages does not mean that malicious nodes have the ability to impersonate honest nodes. Rather, their objective is to judiciously increase the frequency of chosen ids to bias the sample list maintained by nodes. We model malicious behaviors through a strong adversary that fully controls these malicious nodes. More precisely, the adversary model that we consider follows the lines of [8,10]. However, we distinguish between two types of adversary: the *omniscient* adversary that is able to eavesdrop all messages exchanged within the system, and the *blind* adversary that can only observe messages sent or received by malicious nodes.

For both models, we assume that the adversary cannot control more than a fraction $k < 1$ of malicious nodes in the overlay. We also suppose that the adversary can neither drop a message exchanged between two honest nodes nor tamper with its content without being detected. This is achieved by assuming the existence of a signature scheme (and the corresponding public-key infrastructure) ensuring the authenticity and integrity of messages. The signature scheme enables to verify the validity of a signature on a message (*i.e.*, the authenticity and integrity of this message with respect to a particular node). Recipients of a message ignore any message that is not signed properly. Nodes ids and keys (private and public) are acquired via a registration authority [21] and it is assumed that honest nodes never reveal their private keys to other nodes. We also assume the existence of private channels (obtained through cryptographic means) between each pair of nodes preventing an adversary from eavesdropping and unnoticeably tampering with the content of a message exchanged between two honest nodes through this channel. A malicious node has complete control over the messages it sends and receives. Finally, we do not consider Sybil attacks [12], which mainly consist in flooding the system with numerous fake identifiers. We assume the existence of some external mechanism for solving this problem such as an off-line certification authority.

3.3 Sampling Assumptions

Similarly to Bortnikov *et al.* [8], we assume that there exists a time T_0 such that after that moment, the churn of the system ceases. This assumption is necessary to make the notion of uniform sample meaningful. Thus from T_0 onwards, the population of the system $\mathcal{S}$ is composed of $n \ll 2^l$ nodes, such that at least $(1-k)n$ of them are honest and no more than kn of them are malicious, and thus are controlled by the adversary. The subset of honest nodes in the overlay is denoted by $\mathcal{N}$. Finally, we assume that all the nodes in $\mathcal{S}$ are weakly connected from time T_0 onwards, which means that there exists a path between any pair of nodes in $\mathcal{S}$ in the underlying undirected graph whose vertices represent the nodes of $\mathcal{S}$ and edges are the communication links between these nodes.

Notation	Meaning
$\mathcal{S}$	The dynamic system
n	Number of nodes in $\mathcal{S}$
$v_1, \ldots v_n$	Nodes identifiers
$\mathcal{N}$	Subset of honest nodes in $\mathcal{S}$
k	Proportion of malicious nodes in $\mathcal{S}$
σ	Input stream
m	Size of the input stream σ
δ	Number of identifiers injected by the malicious nodes in σ
m_u	Number of occurrences of node identifiers u in σ
T_0	Time after which the churn ceases
$\mathcal{U}$	Uniformity property
$\mathcal{F}$	Freshness property
s	Sampling strategy
T_s	Expecting time required for strategy s to converge
$S^s_{v_i}$	Sampling component run at node v_i implementing strategy s
$S^s_{v_i}(t)$	Output of the sampling component $S^s_{v_i}$ at time t
Γ_{v_i}	Local memory of node v_i
p, q	Probability distributions
$p^{(\mathcal{U})}$	Uniform distribution
$H(p)$	Entropy of p
$H(p, q)$	Cross entropy of p relative to q
$\mathcal{D}(p\|\|q)$	Kullback-Leibler divergence of p relative to q
$\mathcal{D}^{\max}$	Maximum value of $\mathcal{D}(q\|\|p^{(\mathcal{U})})$
τ_s	Robustness threshold of strategy s

Table 1. List of symbols and notations

3.4 Notations

The main symbols and notations used in this article are summarized in Table 1. Most of these notations are formally defined in the following.

4 Sampling Component

4.1 Assumptions and Terminology

Each node $v_i \in \mathcal{N}$ has locally access to a *sampling component*[4] as illustrated in Figure 1. The sampling component implements a *strategy* s and has uniquely access to a local data structure Γ_{v_i}, referred to as the *sampling memory*. The size of the sampling memory Γ_{v_i} is bounded and is denoted by $|\Gamma_{v_i}|$. The sampling component $S^s_{v_i}$ is fed with an infinite stream $< v_1, v_2, \ldots >$ of (possibly non unique) node ids that correspond to the node ids periodically received by node

[4] Although malicious nodes have also access to a sampling component, we cannot impose any assumptions on how they feed it or use it as their behavior can be totally arbitrary.

$v_i \in \mathcal{N}$. This stream results either from the propagation of node ids through gossip-based algorithms (namely through push, or pull or push-pull mechanisms initiated by v_i and its neighbors), or from the node ids received during random walks initiated at v_i, or even resulting from induced churn. The fingerprint of an input stream is a collection of weighted points in which each node id is weighted by the number of times it appears in the stream. Specifically, a stream of node ids can be summarized by $< (v_1, m_{v_1}), \ldots, (v_n, m_{v_n}) >$, where v_i denotes the identifier of a node in $\mathcal{S}$ and $m_{v_i} \in \mathbb{N}$ represents the number of times v_i appears in the stream. At each time t, the following three steps are atomically executed. To begin with, the first element of the stream, say node id v_j, is given as input to the sampling component. Then, the sampling component $S^s_{v_i}$ reads v_j and removes it from the stream. Finally, according to its strategy s, $S^s_{v_i}$ may store or not v_j in Γ_{v_i} and outputs at most one node id.

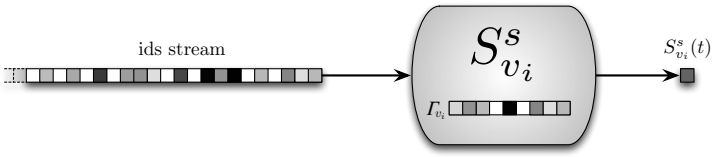

Fig. 1. Sampling component of node $v_i \in \mathcal{N}$.

For example, the strategy s may consist in storing v_j if Γ_{v_i} is not full, or in substituting v_j for a randomly chosen node id that belongs to Γ_{v_i}, or simply in dropping v_j. The output of the sampling component at time t, denoted $S^s_{v_i}(t)$, is chosen among the node ids in Γ_{v_i} according to strategy s. For instance, strategy s may consist in choosing a random node id in Γ_{v_i} [32,33] or the smallest node id under a given min-wise permutation [8]. The maximum finite hitting time needed for the sampling component $S^s_{v_i}$ to reach a uniform sample is denoted by T_s. Clearly, T_s depends on the strategy s implemented by the sampling component and also on the stream of node ids the sampling component is fed with. We assume that the sampling strategy is known by the adversary in the sense that the algorithm used is public knowledge. However, if the algorithm is a randomized one, the adversary does not have access to the local random coins used by the honest nodes.

Finally, δ represents the number of node ids injected by the adversary in the input stream of node u during the time interval T_s. Note that it does not matter whether the injected node ids correspond to the node ids of malicious nodes or not. Indeed, the unique objective of the adversary is to bias the input stream in such a way that whatever the strategy s of the sampler component, its output $S^s_{v_i}(t)$ cannot guarantee both the uniform and freshness properties.

4.2 Sampling Properties

We consider the problem of achieving an unbiased (uniform) and fresh sampling in large scale unstructured peer-to-peer systems subject to adversarial attacks. A strategy s that solves this problem has to meet the following two properties: *i)* Uniformity, which states that any node in the overlay should have the same probability to appear in the sample of honest nodes in the overlay, and *ii)* Freshness, which states that any node should have a non-null probability to appear infinitely often in the sample of any honest node in the overlay. More formally, the sampling strategy s should guarantee the following two properties.

Property 1 (Uniformity). Let $\mathcal{N}$ be a weakly connected graph from time T_0 onwards, then for any time $t \geq T_s$, for any node $v_j \in \mathcal{S}$, and for any node $v_i \in \mathcal{N}$,

$$\mathbb{P}[v_j \in S^s_{v_i}(t)] = \frac{1}{|\mathcal{S}|}.$$

Property 2 (Freshness). Let $\mathcal{N}$ be a weakly connected graph from time T_0 onwards, then for any time $t \geq T_s$, for any node $v_j \in \mathcal{S}$, and for any node $v_i \in \mathcal{N}$,

$$\{t' > t \mid S^s_{v_i}(t') = v_j\} \neq \emptyset \text{ with probability 1.}$$

Note that uniformity by itself does not imply freshness, and *vice versa*. Indeed, the former does not impose any restriction on the freshness of output node ids, while the latter one does not provide any guarantee regarding the equiprobability of node ids to be chosen as samples. Moreover, as each node v_j in $\mathcal{S}$ has a non-null probability to be returned by $S^s_{v_i}$ at time t, v_j must appear at least once in the input stream. Thus, $\forall v_j \in \mathcal{S}$, starting from time T_s, $m_{v_j} > 0$. Note that, as previously mentioned, the model and analysis presented in this paper are independent from the way the stream of node ids at each node v_i is generated.

5 Characterization of the Uniform and Fresh Sampling Problem

We start our characterization by showing that the adversary can bias the input stream in such a way that neither uniform nor freshness properties can be met. This is achieved by flooding the input stream with sufficiently many chosen node ids. Specifically, Lemma 1 states that for any strategy s, if the number δ of non unique node ids that appear in the input stream of node $v_i \in \mathcal{N}$ during T_s time units exceeds a given threshold then it is impossible for any node in the overlay to equally likely appear as a sample of node v_i, and this holds forever. Let $\mathcal{C}_1$ be a condition on the value of δ:

$$\delta \leq T_s - (1-k)|\mathcal{S}|. \qquad (\mathcal{C}_1)$$

Condition $\mathcal{C}_1$ characterizes the fact that for any honest node $v_i \in \mathcal{N}$, during the time interval T_s, v_i has a non-null probability to appear in the input stream. We have the following lemma.

Lemma 1.

$$\neg(\mathcal{C}_1) \Longrightarrow \neg\mathcal{U} \wedge \neg\mathcal{F}.$$

Proof. Let $v_i \in \mathcal{N}$. Suppose that Condition $\mathcal{C}_1$ does not hold, namely it exists an adversarial behavior such that

$$\delta > T_s - (1-k)|\mathcal{S}|.$$

In this case, the number of honest node ids in the input stream at v_i (*i.e.*, $T_s - \delta$) is strictly lower than $(1-k)|\mathcal{S}|$, which means formally that

$$T_s - \delta < (1-k)|\mathcal{S}|.$$

By assumption (*cf.* Section 4.1), the overlay is populated by $(1-k)|\mathcal{S}|$ honest nodes. Thus, as the adversary manages to flood the input stream at v_i, there exists at least one node id $v_j \in \mathcal{S}$ that will never appear in the stream. Therefore, whatever the strategy s, v_i's sampling component can never output v_j. Thus,

$$\forall t > T_0,\ \mathbb{P}[v_j \in S_{v_i}(t)] = 0, \tag{1}$$

which clearly violates Property $\mathcal{U}$.

Equation (1) can be rewritten as $\exists t > T_0, \exists v_j \in \mathcal{S}, \forall t' > t, \mathbb{P}[v_j \in S_{v_i}(t')] = 0$, which has for consequence that the set of instants t' for which v_j can be sampled by v_i is empty. Formally,

$$\mathbb{P}[\{t' | t' > T_0 \wedge v_j \in S_{v_i}(t')\} = \emptyset] > 0,$$

which violates Property $\mathcal{F}$, and completes the proof of the lemma. □

We now assume that Condition $\mathcal{C}_1$ holds. The second lemma states that if the size of the sampling memory is large enough, then whatever the constrained adversarial behavior, the sampling component succeeds in exhibiting uniform and fresh samples. This results in a sufficient condition to solve our problem. Specifically, let $\mathcal{C}_2$ be defined as follows

$$|\Gamma_{v_i}| < |\mathcal{S}|. \tag{$\mathcal{C}_2$}$$

Condition $\mathcal{C}_2$ characterizes the fact that nodes cannot maintain the full knowledge of the population overlay essentially due to scalability reasons. We can now prove the following lemma.

Lemma 2.

$$(\mathcal{C}_1) \wedge \neg(\mathcal{C}_2) \Longrightarrow \mathcal{U} \wedge \mathcal{F}.$$

Proof. Proof of the lemma is straightforward. By Condition $\mathcal{C}_1$, any node $v_j \in \mathcal{S}$ has a non-null probability to appear in the input stream of any node $v_i \in \mathcal{N}$. By assumption of the lemma, $|\Gamma_{v_i}| \geq |\mathcal{S}|$. Consider the basic strategy s of v_i's sampling component that consists in storing into Γ_{v_i}, any new id read from the input stream. Then eventually, all the node ids will be present into Γ_{v_i}, and

thus, according to an uniform selection strategy, any node v_j is equally likely to be chosen in Γ_{v_i}, which guarantees Property $\mathcal{U}$.

Moreover, v_i has the possibility to return infinitely often any node id v_j present in Γ_{v_i}. Thus for any time t, the set of times t' with $t' > t$ at which v_j has been chosen in Γ_{v_i} has a zero probability to be empty, which ensure Property $\mathcal{F}$ and completes the proof. □

The following Lemma completes the characterization of the problem.

Lemma 3.

$$(\mathcal{C}_1) \wedge (\mathcal{C}_2) \Longrightarrow \neg(\mathcal{U} \wedge \mathcal{F}).$$

Proof. Suppose that both Conditions $\mathcal{C}_1$ and $\mathcal{C}_2$ hold. Proving that $\neg(\mathcal{U} \wedge \mathcal{F})$ is equivalent to showing that $(\neg\mathcal{F} \vee \neg\mathcal{U})$ holds, and thus, that $(\mathcal{F} \Longrightarrow \neg\mathcal{U})$ is true. Suppose that $(\mathcal{C}_1) \wedge (\mathcal{C}_2) \wedge \mathcal{F}$ is met, we now show that $\mathcal{U}$ cannot hold.

Consider any node $v_i \in \mathcal{N}$ (the set of honest nodes) and let $\Gamma_{v_i}(t)$ denote the content of v_i's sampling memory at the instant t. From Condition $\mathcal{C}_2$,

$$\forall t' \geq T_0, \exists v_j \in \mathcal{S}, \quad v_j \notin \Gamma_{v_i}(t'). \tag{2}$$

In particular, Equation (2) is true for $t' = T_s$. Let node $v_j \in \mathcal{S}$ be such that $v_j \notin \Gamma_{v_i}(T_s)$, then by assumption Property $\mathcal{F}$ holds. Thus

$$\exists t > T_s, \quad v_j \notin \Gamma_{v_i}(T_s) \wedge v_j \in \Gamma_{v_i}(t). \tag{3}$$

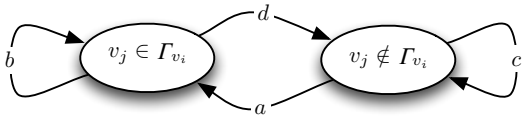

Fig. 2. Markov chain representing the evolution of v_j's presence in the sampling memory Γ_{v_i} of node $v_i \in \mathcal{N}$.

The presence of a node id in the local memory of the sampling component can be represented as a Markov chain. Figure 2 depicts the evolution of $v_j \in \Gamma_{v_i}$ as a function of time. Labels a, b, c and d on the edges represent the probability of transitions from both states such that we have $a + c = b + d = 1$. From Equation (3), we have $a > 0$ and thus, $c < 1$. We prove by contradiction that $d > 0$.

Suppose that $d = 0$, then $\forall t'' \geq t$, $v_j \in \Gamma_{v_i}(t'')$, the state $(v_j \in \Gamma_{v_i})$ is absorbing. Consider an overlay that contains only two nodes, v_i and v_j. By assumption, at least one of the two nodes is honest ($k < 1$). Let us assume that v_i is honest (the proof is similar for v_j). Then, by Condition $\mathcal{C}_2$, we have $|\Gamma_{v_i}| = 1$ (the case $|\Gamma_{v_i}| = 0$ trivially leads to an impossibility). By assumption, we have

$\forall t'' \geq t,\ v_j \in \Gamma_{v_i}(t'')$ and as $|\Gamma_{v_i}| = 1$, we also have $\forall t'' \geq t,\ \Gamma_{v_i}(t'') = \{v_j\}$. Consequently, whatever the strategy s implemented in v_i's sampling component,

$$\forall t'' \geq t,\ \mathbb{P}[v_i \in S^s_{v_i}(t'')] = 0 \quad \Longrightarrow \quad \mathbb{P}[\{t'' | t'' > t \wedge v_i \in S^s_{v_i}(t'')\} = \emptyset] > 0,$$

which contradicts $\mathcal{F}$ and also the assumption of the lemma. Thus $d > 0$ and, *a fortiori*, $b < 1$, and no state is absorbing.

Suppose now that $\mathcal{U}$ holds, we now prove the lemma by contradiction. Consider once again the situation in which the overlay is populated by only two nodes, v_i and v_j. As previously, suppose that node v_i is honest and that $|\Gamma_{v_i}| = 1$. The evolution of the sampling memory at node v_i can be modeled by a Markov chain as represented in Figure 3. By assumption, $\mathcal{F}$ holds, and thus infinitely often and successively, both v_i and v_j appear in Γ_v. Moreover by assumption, $\mathcal{U}$ also holds, that is, $\forall t \geq T_s, \mathbb{P}[v_j \in S^s_{v_i}(t)] = \mathbb{P}[v_i \in S^s_{v_i}(t)] = \frac{1}{2}$. As a consequence, v_j has the same probability as v_i to be in Γ_v, whatever the number of times v_j and v_i appear in the stream before time T_s.

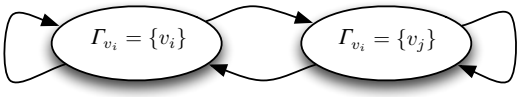

Fig. 3. Markov chain representing the state of the local memory Γ_{v_i} of v_i.

Suppose now that node v_j is malicious. By Condition $\mathcal{C}_1$, node id v_j can appear in v_i's stream no more than $T_s - 1$ times during any sliding window of T_s time units. As $|\Gamma_{v_i}| = 1$, only a single node id can be stored, and beyond this node id, no other additional information can be stored. We now demonstrate that whatever the strategies s implemented by v_i's sampling component, they all lead to a contradiction.

Blind replacement. At any time t, the sampling component reads the first node id in the stream, and stores it in Γ_{v_i} in place of the previous one. By construction, any strategy has to select its output among the elements stored in Γ_{v_i}, thus the output of the sampling component follows the same probability distribution as the one observed in the stream. As the adversary can flood the stream with up to $T_s - 1$ malicious node ids, this means that Property $\mathcal{U}$ cannot be met.

No replacement. Similarly to the blind replacement strategy, node ids are read from the stream and stored in Γ_{v_i} up to time t, in which t is the first time at which a specific node id is read. From time t onwards, this specific node id is kept in Γ_{v_i}, independently from the node ids read from the stream after t, leading to an absorbing state of the Markov chain. For instance, this specific node id can be the smallest image value under a random min-wise

independent function such as the min-wise permutation [8]. Clearly, this strategy violates Property $\mathcal{F}$.

Probabilistic replacement. This strategy consists in substituting the current node id in Γ_{v_i} with the next one read from the stream according to a given probability law. To guarantee that $\forall t, \mathbb{P}[v_j \in S^s_{v_i}(t)] = \mathbb{P}[v_i \in S^s_{v_i}(t)] = \frac{1}{2}$, then either both v_i and v_j have an equal probability to appear in the stream or the sampling component must be able to remember the node ids it has seen in the past to guarantee that, at any time t, each node id has the same probability to be chosen as sample. The former case does not hold as by assumption, the adversary can flood the stream with up to $T_s - 1$ malicious ids. Moreover, the latter case is impossible as by assumption $|\Gamma_{v_i} = 1|$. Therefore, only a single information can be stored (*e.g.*, it is impossible to store both a node id and a counter), and as consequence Property $\mathcal{U}$ cannot hold.

Thus $(\mathcal{C}_1) \wedge (\mathcal{C}_2) \Longrightarrow \neg(\mathcal{U} \wedge \mathcal{F})$, which concludes the proof of the lemma. □

The last lemma, described below, reformulates the necessary condition of the problem characterization by combining Lemmata 1 and 3.

Lemma 4.

$$\mathcal{U} \wedge \mathcal{F} \Longrightarrow (\mathcal{C}_1) \wedge \neg(\mathcal{C}_2).$$

Proof. The contrapositive form of writing Lemma 3 is $\mathcal{U} \wedge \mathcal{F} \Longrightarrow \neg((\mathcal{C}_1) \wedge (\mathcal{C}_2))$, and thus by distribution,

$$\mathcal{U} \wedge \mathcal{F} \Longrightarrow \neg(\mathcal{C}_1) \vee \neg(\mathcal{C}_2). \tag{4}$$

On the other hand, the contraposition of Lemma 1 leads to $\mathcal{U} \vee \mathcal{F} \Longrightarrow (\mathcal{C}_1)$. As $(\mathcal{U} \wedge \mathcal{F} \Rightarrow \mathcal{U} \vee \mathcal{F})$, we have

$$\mathcal{U} \wedge \mathcal{F} \Longrightarrow (\mathcal{C}_1). \tag{5}$$

By combining Equations 4 and 5, the following relation holds

$$\mathcal{U} \wedge \mathcal{F} \Longrightarrow (\mathcal{C}_1) \wedge (\neg(\mathcal{C}_1) \vee \neg(\mathcal{C}_2)),$$

which is equivalent to

$$\mathcal{U} \wedge \mathcal{F} \Longrightarrow ((\mathcal{C}_1) \wedge \neg(\mathcal{C}_1)) \vee ((\mathcal{C}_1) \wedge \neg(\mathcal{C}_2)).$$

By contradiction, $(\mathcal{C}_1) \wedge \neg(\mathcal{C}_1)$ cannot hold, leading to

$$\mathcal{U} \wedge \mathcal{F} \Longrightarrow (\mathcal{C}_1) \wedge \neg(\mathcal{C}_2),$$

which completes the proof. □

The uniform and fresh sampling problem defined in Sections 3 and 4 is completely characterized by the following theorem.

Theorem 1. *$(\mathcal{C}_1) \wedge \neg(\mathcal{C}_2)$ is a necessary and sufficient condition for the uniform and freshness properties to hold.*

Proof. This result follows directly from the statements of Lemma 2 and 4. □

Given this coarse-grained characterization, it appears to be unreasonable to ensure such conditions in real life applications. To address this issue, the following sections provide some finer-grained characterization of sampling strategies. These strategies are capable of unbiasing an input stream according to the value of norm of the entropy of this stream.

6 Information Divergence of Data Streams

A natural approach to detect changes on data streams is to model it as a distribution and to compute the distance between the observed stream and the ideal one. The metric we use in our context is the Kullback-Leibler (KL) divergence, also sometimes called the relative entropy [37].

Definition 1 (Kullback-Leibler divergence).
Given two probability distributions $p = \{p_1, \ldots, p_n\}$ and $q = \{q_1, \ldots, q_n\}$, the Kullback-Leibler divergence of p_i relative to q_i is defined as the expected value of the likelihood ratio with respect to q_i. Specifically,

$$\mathcal{D}(p||q) = \sum_{i=1}^{n} p_i \log_2 \frac{p_i}{q_i} = H(p,q) - H(p), \tag{6}$$

where $H(p) = -\sum p_i \log_2 p_i$ is the (Shannon) entropy of p and $H(p,q) = -\sum p_i \log_2 q_i$ is the cross entropy of p and q (by convention, $0 \log_2 0 = 0$).

The KL-divergence is a member of a larger class of distances known as the Ali-Silvey distances [38]. For the sake of clarity, we will use the notation log to denote the logarithm in base 2. Let $p^{(\mathcal{U})}$ be the uniform distribution corresponding to a uniform stream (*i.e.*, $\forall i \in [1..n], p_i^{(\mathcal{U})} = \frac{1}{n}$), and q be the probability distribution corresponding to the input stream. Following the classical use of the KL-divergence, we consider $\mathcal{D}(q||p^{(\mathcal{U})})$ as a measure of the divergence of the stream from the ideal one. Note that a probabilistic algorithm computing on the fly the KL divergence of any stream with respect to an ideal one has been proposed in [39]. Experimental results have shown that the estimation provided by this algorithm remains accurate even for different adversarial settings in which the quality of other methods dramatically decreases.

Definition 2 (τ-closeness). *A stream of node ids σ is τ-close if the KL-divergence between the probability distribution q corresponding to σ and the uniform probability distribution $p^{(\mathcal{U})}$ is below or equal to a given value τ called the robustness threshold, in whch τ is a positive real value.*

Finally given probability distributions, the Earth Mover's Distance (EMD) [40] measures the minimal amount of work needed to transform one distribution to another by moving the probability mass between events. We rely on this metric to quantify the effort that an adversary exerts to bias the input stream. In our context, a unit of work corresponds to dropping one id and to pushing another id instead in the input stream.

7 Omniscient Adversary Model

In this section, we study the behavior of an omniscient adversary, which has the capacity to eavesdrop on all the messages sent and received by all the nodes in $\mathcal{S}$. We demonstrate that the strategy pushing all the probability mass over a single id is the one maximizing the bias of the input stream so that it becomes far from the uniform distribution. We also describe an optimal strategy achieving it.

In the following, the analysis of infinite input stream is restricted to any window of length T_s observed from time $t \geq T_0$. As previously said, T_s depends on the sampling strategy and thus can be arbitrarily large. For the sake of simplicity, the term "stream" will denote in the remaining of the paper the stream restricted to any such window of length T_s.

Let $\overline{\sigma}$ be a stream such that the id of each node in $\mathcal{S}$ appears exactly once in the stream, except for a unique id that appears in all the remaining slots. Therefore, there exists a unique $v_i \in \mathcal{S}$ such that $m_{v_i} = T_s - (n-1)$ and $\forall v_j \neq v_i \in \mathcal{S}, m_{v_j} = 1$. The following theorem states that the probability distribution associated to this particular stream is the one that has the maximal divergence from the uniform distribution.

Theorem 2. (Maximal divergence from the uniform distribution) *Let $p^{(\mathcal{U})}$ be the uniform distribution corresponding to a uniform stream, that is, $\forall i \in [1..n], p_i^{(\mathcal{U})} = \frac{1}{n}$, and $\overline{q}$ be the probability distribution corresponding to $\overline{\sigma}$ (*i.e.*, it exists a unique $v_i \in \mathcal{S}, \overline{q}_{v_i} = \frac{T_s-(n-1)}{T_s}$ and $\forall v_j \in \mathcal{S}, v_j \neq v_i \Rightarrow \overline{q}_{v_j} = \frac{1}{T_s}\Big)$. Then, for any possible probability distribution q,*

$$\mathcal{D}(q||p^{(\mathcal{U})}) \leq \mathcal{D}(\overline{q}||p^{(\mathcal{U})}).$$

Proof. Let q be the probability distribution representing any valid input stream on $(T_0, T_s]$. We have $\forall v_i \in \mathcal{S}, q_{v_i} = \frac{m_{v_i}}{T_s}$, where m_{v_i} is the number of times v_i is present in the input stream. It follows that

$$\begin{aligned}\mathcal{D}(q||p^{(\mathcal{U})}) &= H(q, p^{(\mathcal{U})}) - H(q) \\ &= -\sum_{i=1}^{n} q_i \log\left(p_i^{(\mathcal{U})}\right) - H(q) = \log(n) - H(q).\end{aligned}$$

Therefore, maximizing $D(q||p^{(\mathcal{U})})$ amounts to minimizing $H(q)$, which is equivalent to maximize $\sum_{i=1}^{n} m_{v_i} \log\left(\frac{m_{v_i}}{T_s}\right)$. We characterize the stream that minimizes $H(q)$ under the following constraints:

$$\begin{cases} 1 \leq m_{v_i} \leq T_s \ \text{ with } 1 \leq i \leq n, \\ \sum_{i=1}^{n} m_{v_i} = T_s. \end{cases} \tag{7}$$

From these constraints, we immediately have $1 \leq m_{v_i} \leq T_s - (n-1)$. To relax the second constraint, we pose $m_{v_n} = T_s - \sum_{i=1}^{n-1} m_{v_i}$. Let function f be such that

$$f(m_{v_1}, \ldots, m_{v_{n-1}}) = \sum_{i=1}^{n-1} m_{v_i} \log\left(\frac{m_{v_i}}{T_s}\right) + \left(T_s - \sum_{i=1}^{n-1} m_{v_i}\right) \log\left(1 - \sum_{i=1}^{n-1} \frac{m_{v_i}}{T_s}\right).$$

Function f is differentiable on its domain $\mathcal{I}_s = [1..T_s - n + 1]^{n-1}$, thus we get

$$\begin{aligned}\frac{\mathrm{d}f}{\mathrm{d}m_{v_j}}(m_{v_1}, \ldots, m_{v_{n-1}}) &= \log\left(\frac{m_{v_j}}{T_s}\right) + m_{v_j}\frac{T_s}{m_{v_j}} + \log\left(1 - \sum_{i=1}^{n-1}\frac{m_{v_i}}{T_s}\right) \\ &\quad + \frac{T_s - \sum_{i=1}^{n-1} m_{v_i}}{1 - \sum_{i=1}^{n-1}\frac{m_{v_i}}{T_s}} \\ &= \log\left(m_{v_j}\right) + \log\left(T_s - \sum_{i=1}^{n-1} m_{v_j}\right) + 2(T_s - \log(T_s)).\end{aligned}$$

According to Equation 7, we have

$$\log\left(m_{v_j}\right) + \log\left(T_s - \sum_{i=1}^{n-1} m_{v_j}\right) \geq 0$$

and, as $T_s \gg 1$, this implies $T_s - \log(T_s) > 0$. Then, we obtain that $\frac{\mathrm{d}f}{\mathrm{d}m_{v_j}} > 0$, leading to the fact that f is strictly increasing according to m_{v_j}. The maximum is then reach for $m_{v_j} = T_s - n + 1$ as f is a Schur-convex function.

From this set of constraints (*cf.* Equation 7), if the maximum of $\mathcal{D}(q||p^{(\mathcal{U})})$ is reached for $m_{v_j} = T_s - n + 1$ then $\sum_{i=1, i\neq j}^{n} m_{v_i} = n - 1$ implies that $\forall i \in [1..n], i \neq j, m_{v_i} = 1$, which concludes the proof. □

Theorem 2 enables us to formulate an upper-bound $\mathcal{D}^{\max}$ on the KL-divergence between the uniform stream and any other stream:

$$\mathcal{D}^{\max} = \mathcal{D}(\overline{q}||p^{(\mathcal{U})}) = \log(n) + \log(T_s) - \left(1 - \frac{n-1}{T_s}\right)\log\left(T_s - n + 1\right). \quad (8)$$

As a consequence, any input stream σ is $\mathcal{D}^{\max}$-close (*cf.* Definition 2).

To determine the minimal effort that the adversary has to exert to bias the input stream so that both uniformity and freshness properties do not hold, we use the Earth Mover's Distance (EMD) between the uniform distribution and the target one. In the following, when we say that the adversary replaces node id v_i by node id v_j, we mean that he drops v_i from the input stream and injects v_j instead. Recall that the adversary is able to drop and inject node ids only from the nodes it controls. However, as mention in the introduction, the adversary may succeed in surrounding a honest node with malicious nodes so that it may shape by itself the input stream of this honest node.

Lemma 5. (Optimal strategy to maximize the divergence)
Given an input stream σ, replacing the less frequent node id in σ with the most frequent one maximizes the gain in KL-divergence with respect to the uniform distribution for the same amount of work as measured by the EMD distance.

Proof. Given an input stream σ represented by the probability distribution q, we construct the input stream σ' from σ by substituting one occurrence of node id v_i with node id v_j so that $\mathcal{D}(q'||p^{(\mathcal{U})})$ is maximized after this replacement (in which q' denotes the probability distribution representing σ'). This amounts to maximize $\left[\mathcal{D}(q'||p^{(\mathcal{U})}) - \mathcal{D}(q||p^{(\mathcal{U})})\right]$. Recall that all node ids in $\mathcal{S}$ must be present in σ'. Therefore, we search for the node id pair (v_i, v_j) such that

$$\begin{cases} m'_{v_j} = m_{v_j} + 1 \\ m'_{v_i} = m_{v_i} - 1 \\ v_j = \arg\max_{v_j \in \mathcal{S}} \left(q'_{v_j} \log\left(q'_{v_j}\right) - q_{v_j} \log\left(q_{v_j}\right)\right) \\ v_i = \arg\max_{v_i \in \mathcal{S}} \left(q'_{v_i} \log\left(q'_{v_i}\right) - q_{v_i} \log\left(q_{v_i}\right)\right) \end{cases}$$

Consider the function $f : x \mapsto x\log(x)$, which is strictly increasing. For any $k \in \mathcal{S}$, we have $q_{v_k} = m_{v_k}/T_s$. Thus,

$$\begin{cases} v_j = \arg\max_{v_j \in \mathcal{S}} \left(f\left(\frac{m_{v_j}+1}{T_s}\right) - f\left(\frac{m_{v_j}}{T_s}\right)\right) \\ v_i = \arg\max_{v_i \in \mathcal{S}} \left(f\left(\frac{m_{v_i}-1}{T_s}\right) - f\left(\frac{m_{v_i}}{T_s}\right)\right) \end{cases}$$

$$\Longrightarrow \begin{cases} v_j = \arg\max_{v_j \in \mathcal{S}} \ m_{v_j} \\ v_i = \arg\min_{v_i \in \mathcal{S}} \ m_{v_i} \end{cases}$$

This derives from the fact that the function $g_1 : x \mapsto f(\frac{x+1}{T_s}) - f(\frac{x}{T_s})$ (respectively $g_2 : x \mapsto f(\frac{x-1}{T_s}) - f(\frac{x}{T_s})$) is strictly increasing (respectively strictly decreasing). The optimal node id replacement maximizing the KL-divergence gain is reached by replacing the less frequent node id v_i with the most frequent one v_j. □

Algorithm 1 (run by the adversary) shows an optimal implementation of Lemma 5 with respect to the number of performed replacements. Specifically, the inputs of the algorithm are an input stream σ and the robustness threshold τ_s. Recall that τ_s is the robustness threshold of the sampling strategy s implemented by S_u^s, which means that for any τ_s-close input stream σ, the sampling strategy s is able to output a uniform and fresh sample. The goal of the greedy Algorithm 1 is to tamper with the input stream σ in order to increase its KL-divergence above τ_s with a minimum effort.

By assumption, the adversary is omniscient and therefore has the capacity to observe the entire input stream σ. From Section 4, the adversary knows the strategy s of the sampler, and thus can compute the value of τ_s. The value of the maximum divergence $\mathcal{D}^{\max}$ is computed using Relation (8). If $\mathcal{D}^{\max}$ is larger than or equal to the robustness threshold, the algorithm returns "fail". Otherwise at each iteration, the adversary performs the optimal node id replacement until the KL-divergence exceeds the robustness threshold. Remember however, that

Algorithm 1: Adversary biasing strategy

Data: an input stream σ, the robustness threshold τ_s
Result: the number of replacements ℓ if it exists

```
1  if τ_s ≥ D^max then
2      return "fail"
3  else
4      ℓ ← 0;
5      v_j ← arg max_{v_j ∈ S} m_{v_j};
6      while ( D(q_σ || p^(U)) ≤ τ_s ) do
7          v_i ← arg min_{ {v ∈ S : m_{v_i} ≠ 1} } m_{v_i};
8          let k be the index of an item in the part of the stream controlled by an
           adversary such that σ[k] = v_i ;
9          σ[k] ← v_j   //one occurrence of v_i is dropped and v_j is injected instead;
10         ℓ ← ℓ + 1;
11     return ℓ
```

the adversary cannot drop messages that have been sent or forwarded by nodes it does not control (*i.e.*, the honest ones). Note that at Lines (8) and (9) of Algorithm 1 both m_{v_i} and m_{v_j} are updated. Counter ℓ returned by Algorithm 1 represents the number of replacements done by the adversary.

Consider a sampling strategy s, its robustness threshold τ_s, and an input stream σ. Let ℓ be the number of replacements executed by Algorithm 1. If we denote by $q_{\sigma(\ell)}$ the probability distribution derived from σ after these ℓ optimal replacements, we have the following corollary.

Corollary 1. (Lower bound on the effort exerted by an omniscient adversary)
The minimum number of replacements an omniscient adversary has to apply to exceed τ_s is

$$\delta = \inf \left\{ \ell \in \mathbb{N} : \mathcal{D}(q_{\sigma(\ell)} || p^{(\mathcal{U})}) > \tau_s \right\}. \tag{9}$$

8 Blind Adversary Model

In this section, we study the behavior of a blind adversary, that is an adversary that only has the capacity to observe messages sent or received by the nodes he controls. A strategy that the adversary might apply to bias the input stream is to choose a node id (possibly one that belongs to a malicious node but not necessarily) and to push it in the input stream as much as possible. We show that this strategy is optimal with respect to the effort exerted by the adversary and we derive a lower bound on the expected minimum amount of work a blind adversary has to exert to bias the input stream.

Theorem 3. (Lower bound on the expected effort exerted by a blind adversary)

Let s be a sampling strategy, τ_s its robustness threshold and T_s its maximum convergence time. The minimum number of replacements a blind adversary has to apply in expectation to exceed τ_s is given when the input stream is the uniform one. We have

$$\tilde{\delta} = \inf \{\ell \in \mathcal{I}_s : \mathcal{R}_\ell > \tau_s\} \tag{10}$$

$$\begin{aligned} \textit{where } \mathcal{I}_s = & \left[0..T_s - n + 1 - \left\lfloor \frac{T_s}{n} \right\rfloor\right] \\ \mathcal{R}_\ell = & \frac{1}{n}\left(\left\lfloor \frac{\ell - 1}{\lfloor \frac{T_s}{n} - 1 \rfloor} \right\rfloor \log\left(\frac{T_s}{n}\right) + \log\left(\frac{T_s^2}{T_s + n\ell}\right)\right. \\ & \left. - \log\left(T_s - n\left(1 + (\ell - 1) \bmod \left\lfloor \frac{T_s}{n} - 1 \right\rfloor\right)\right)\right) \end{aligned}$$

Proof. Let us consider the uniform node ids stream on a window of length T_s. For any $v_i \in \mathcal{S}$, v_i is present in the stream T_s/n times in average. The probability distribution $p^{(\mathcal{U})}$ is such that $\forall v_i \in \mathcal{S}, p_{v_i}^{(\mathcal{U})} = 1/n$. From Section 7, we have seen that the optimal strategy for the adversary to bias an input stream is to replace the less frequent node id in this stream with the most frequent one. By assumption, the adversary is blind and cannot observe all the node ids of the input stream. The strategy of the adversary consists in choosing a specific node id v_j and repeatedly pushing v_j in the input stream. Let σ be an input stream and σ' be the stream obtained from σ after one step of this adversarial strategy (*i.e.*, replacing v_i by v_j for some $v_i \in \mathcal{S}$). We have

$$\mathcal{D}(q_{\sigma'}||p^{(\mathcal{U})}) - \mathcal{D}(q_\sigma||p^{(\mathcal{U})}) = \frac{1}{n}\left(\log\left(\frac{m_{v_j}}{m_{v_j} + 1}\right) + \log\left(\frac{m_{v_i}}{m_{v_i} - 1}\right)\right), \tag{11}$$

in which q_σ and $q_{\sigma'}$ represent respectively the probability distributions of σ and σ'. In the following, $q_{\sigma(\ell)}$ denotes the probability distribution derived from σ after ℓ replacements. Given a sampling strategy s, we prove by induction on the number of optimal replacements ℓ that, starting from a uniform stream, the maximum KL-divergence after ℓ replacements is given by $\mathcal{D}(q_{\sigma(\ell)}||p^{(\mathcal{U})}) = \mathcal{R}_\ell$, in which

$$\begin{aligned} \mathcal{R}_\ell = & \frac{1}{n}\left(\left\lfloor \frac{\ell - 1}{\lfloor \frac{T_s}{n} - 1 \rfloor} \right\rfloor \log\left(\frac{T_s}{n}\right) + \log\left(\frac{T_s^2}{T_s + n\ell}\right)\right. \\ & \left. - \log\left(T_s - n\left(1 + (\ell - 1) \bmod \left\lfloor \frac{T_s}{n} - 1 \right\rfloor\right)\right)\right). \end{aligned} \tag{12}$$

Note that ℓ cannot be greater than $\left(T_s - n + 1 - \left\lfloor \frac{T_s}{n} \right\rfloor\right)$. Indeed, all node ids in the initial uniform stream are present at least $\left\lfloor \frac{T_s}{n} \right\rfloor$ times and the maximum number of times a unique id can appear in the stream is $(T_s - n + 1)$. For $\ell = 1$, the claim follows immediately from Equation 11. Now, assume that the claim also

holds for all $1 \leq j \leq \ell$. We show that it holds for $j = \ell + 1$. The KL-divergence with respect to the uniform stream after $\ell + 1$ steps is

$$\mathcal{D}(q_{\sigma(\ell+1)}||p^{(\mathcal{U})}) = \mathcal{D}(q_{\sigma(\ell)}||p^{(\mathcal{U})}) + \mathcal{D}(q_{\sigma(\ell+1)}||p^{(\mathcal{U})}) - \mathcal{D}(q_{\sigma(\ell)}||p^{(\mathcal{U})}). \quad (13)$$

The term $\mathcal{D}(q_{\sigma(\ell+1)}||p^{(\mathcal{U})}) - \mathcal{D}(q_{\sigma(\ell)}||p^{(\mathcal{U})})$ represents the gain of step $(\ell + 1)$, and $\mathcal{D}(q_{\sigma(\ell)}||p^{(\mathcal{U})})$ is given by Equation 12. Two sub-cases need to be considered: *(i)* $\ell \bmod \left\lfloor \frac{T_s}{n} - 1 \right\rfloor \neq 0$ and *(ii)* $\ell \bmod \left\lfloor \frac{T_s}{n} - 1 \right\rfloor = 0$. Case (i): the less frequent node id v_i in the stream at step $\ell + 1$ is the same as the one removed at step ℓ. After ℓ steps, $m_{v_j} = \frac{T_s}{n} + \ell$ and $m_{v_i} = \frac{T_s}{n} - \left(1 + (\ell - 1) \bmod \left\lfloor \frac{T_s}{n} - 1 \right\rfloor\right)$, the right part of Equation 11 is equal to

$$\begin{aligned}
&\frac{1}{n}\left(\log\left(\frac{\frac{T_s}{n} + \ell}{\frac{T_s}{n} + \ell + 1}\right) + \log\left(\frac{\frac{T_s}{n} - \left(1 + (\ell - 1) \bmod \left\lfloor \frac{T_s}{n} - 1 \right\rfloor\right)}{\frac{T_s}{n} - \left(1 + (\ell - 1) \bmod \left\lfloor \frac{T_s}{n} - 1 \right\rfloor\right) - 1}\right)\right) \\
&= \frac{1}{n}\left(\log\left(T_s - n\left(1 + (\ell - 1) \bmod \left\lfloor \frac{T_s}{n} - 1 \right\rfloor\right)\right)\right. \\
&\qquad + \log\left(T_s + n\ell\right) - \log\left(T_s + n(\ell + 1)\right) \\
&\qquad \left. - \log\left(T_s - n\left(2 + (\ell - 1) \bmod \left\lfloor \frac{T_s}{n} - 1 \right\rfloor\right)\right)\right).
\end{aligned}$$

By assumption *(i)*, we have $\left\lfloor \frac{\ell - 1}{\left\lfloor \frac{T_s}{n} - 1 \right\rfloor} \right\rfloor = \left\lfloor \frac{\ell}{\left\lfloor \frac{T_s}{n} - 1 \right\rfloor} \right\rfloor$ and

$$\left(1 + (\ell - 1) \mod \left\lfloor \frac{T_s}{n} - 1 \right\rfloor\right) = \left(\ell \bmod \left\lfloor \frac{T_s}{n} - 1 \right\rfloor\right).$$

From Equation 13, we get

$$\begin{aligned}
\mathcal{D}(q_{\sigma(\ell+1)}||p^{(\mathcal{U})}) = \frac{1}{n}\left(\left\lfloor \frac{\ell}{\left\lfloor \frac{T_s}{n} - 1 \right\rfloor} \right\rfloor \log\left(\frac{T_s}{n}\right) + \log\left(\frac{T_s^2}{T_s + n(\ell + 1)}\right)\right. \\
\left. - \log\left(T_s - n\left(1 + \ell \mod \left\lfloor \frac{T_s}{n} - 1 \right\rfloor\right)\right)\right),
\end{aligned}$$

which ends Case (i). Case (ii). The argumentation is the same as above. However, as $\ell \mod \left\lfloor \frac{T_s}{n} - 1 \right\rfloor = 0$, the node id that has been previously replaced is now present exactly once in the stream. Thus, the adversary needs to randomly choose another node id in the stream before processing the next step of his strategy. Applying Equation 11 at step $\ell + 1$ gives

$$\mathcal{D}(q_{\sigma(\ell+1)}||p^{(\mathcal{U})}) - \mathcal{D}(q_{\sigma(\ell)}||p^{(\mathcal{U})}) = \frac{\left(\log\left(\frac{\frac{T_s}{n} + \ell}{\frac{T_s}{n} + \ell + 1}\right) + \log\left(\frac{\frac{T_s}{n}}{\frac{T_s}{n} - 1}\right)\right)}{n}. \quad (14)$$

By assumption $\left((\ell - 1) \bmod \left\lfloor \frac{T_s}{n} - 1 \right\rfloor = \left\lfloor \frac{T_s}{n} - 1 \right\rfloor - 1\right)$, and by combining the induction hypothesis (Equation 12) with the gain obtained at step $\ell + 1$ (Equa-

tion 14), we get

$$\mathcal{D}(q_{\sigma(\ell+1)}||p^{(\mathcal{U})}) = \frac{1}{n}(3\log(T_s) + \left\lfloor \frac{\ell-1}{\lfloor \frac{T_s}{n} - 1 \rfloor} \right\rfloor \log\left(\frac{T_s}{n}\right) - \log(T_s + n(\ell+1)) - \log(T_s - n) - \log(n)).$$

By assumption of the case $\left\lfloor \frac{\ell}{\lfloor \frac{T_s}{n} - 1 \rfloor} \right\rfloor = \left\lfloor \frac{\ell-1}{\lfloor \frac{T_s}{n} - 1 \rfloor} \right\rfloor + 1$, which proves the induction:

$$\mathcal{D}(q_{\sigma(\ell+1)}||p^{(\mathcal{U})}) = \frac{1}{n}\left(\left\lfloor \frac{\ell}{\lfloor \frac{T_s}{n} - 1 \rfloor} \right\rfloor \log\left(\frac{T_s}{n}\right) + \log\left(\frac{T_s^2}{T_s + n(\ell+1)}\right) - \log\left(T_s - n\left(1 + \ell \mod \left\lfloor \frac{T_s}{n} - 1 \right\rfloor\right)\right)\right).$$

As a conclusion, any value of ℓ that allows the adversary to exceed the robustness threshold τ_s defeats the sampling strategy. Thus, the minimum number of replacement operations $\tilde{\delta}$ is the lower bound of this set of values. □

We now evaluate the minimum amount of work a blind adversary has to exert in the worst case to bias the input stream. In the worst case, the node id v_i the adversary has chosen to blindly flood might be initially present only once in the input stream. In order to bias the input stream, the adversary needs to push id v_i sufficiently often so that the probability of appearance of id v_i reaches the uniform value, with respect to all the other node ids, and then to continue to push this id $\tilde{\delta}$ times so that the divergence between the resulting stream and the uniform one is maximum.

Theorem 4. (Lower bound on the worst-case effort exerted by a blind adversary)
Let s be a sampling strategy, τ_s its robustness threshold and T_s the maximum convergence time of s. The minimum number of replacements the adversary has to apply on a stream in the worst case to exceed τ_s is

$$\tilde{\delta} + \left\lceil \frac{T_s}{n} \right\rceil - 1.$$

Proof. The proof is immediate. First, the adversary has to raise the chosen id at least up to the uniform value. In the worst case, this id is present only once in the initial stream, which costs $\lceil \frac{T_s}{n} \rceil - 1$ replacements to reach a number of occurrences equals to $\lceil \frac{T_s}{n} \rceil$. Moreover, once this id is present in the modified stream $\lceil \frac{T_s}{n} \rceil$ times, the adversary follows the same strategy as before, which requires $\tilde{\delta}$ more steps to guarantee that the robustness threshold τ_s is exceeded. Note that this value is a worst-case bound and not the exact minimum value with respect to τ_s because after the first $(\lceil \frac{T_s}{n} \rceil - 1)$ steps, the modified stream could be different from the uniform one. In this situation, the KL-divergence to the uniform stream is strictly greater than 0, reducing accordingly the amount of work of the adversary to exceed τ_s. □

9 Conclusion and open issues

In this paper, we have focused on the problem of achieving uniform and fresh node sampling in large scale open systems potentially populated with malicious nodes. The node sampling problem consists in guaranteeing that each honest node can maintain a uniform and fresh sample of the whole population of the system. First, we have shown that if the system cannot limit the resources of the adversary, then solving either uniform sampling or fresh sampling is impossible. Then, we have demonstrated that, in the unrealistic setting in which honest nodes have access to a very large memory (*i.e.*, proportional to the size of the system), then the problem becomes trivially solvable. Unfortunately, we have proven that both conditions are necessary and sufficient ingredients to solve the uniform and fresh sampling problem in adversarial environments. Clearly, these strong conditions highlight the damage that adversarial behavior can cause in large-scale unstructured systems.

By modeling input streams as probability distributions, we have also characterized the minimum effort (measured in terms of node ids replacements) that an omniscient and a blind adversary have to exert on the input stream of node identifiers to exceed the robustness threshold that quantifies the power of a sampling strategy. We believe that uniform node sampling should be regarded as a necessary building block to derive larger classes of sampling schemes. This building block is of utmost importance in systems in which the population is continuously evolving and in which it is impossible to capture the full complexity of the network through global snapshots.

References

1. Lv, Q., Cao, P., Cohen, E., Li, K., Shenker, S.: Search and Replication in Unstructured Peer-to-Peer Networks. In: Proceedings of the International Conference on Supercomputing (ICS). (2002)
2. Bertier, M., Busnel, Y., Kermarrec, A.M.: On Gossip and Populations. In: Proceedings of the 16th International Colloquium on Structural Information and Communication Complexity (SIROCCO). (2009)
3. Bollobás, B.: Random Graphs – 2nd Edition. Cambridge University Press (2001)
4. Demers, A., Greene, D., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, H., Swinehart, D., Terry, D.: Epidemic algorithms for replicated database management. In: Proceedings of the 6th ACM Symposium on Principles of Distributed Computing (PODC). (1987)
5. Massoulié, L., Merrer, E.L., Kermarrec, A.M., Ganesh, A.: Peer Counting and Sampling in Overlay Networks: Random Walk Methods. In: Proceedings of the 25th Annual Symposium on Principles of Distributed Computing (PODC), ACM Press (2006) 123–132
6. Stutzbach, D., Rejaie, R., Duffield, N., Sen, S., Willinger, W.: On Unbiased Sampling for Unstructured Peer-to-Peer Networks. IEEE/ACM Transactions on Networking **17**(02) (2009) 377–390
7. Sit, E., Morris, R.: Security Considerations for Peer-to-Peer Distributed Hash Tables. In: Proceedings of the 1rst International Workshop on Peer-to-Peer Systems (IPTPS), Springer-Verlag (2002)

8. Bortnikov, E., Gurevich, M., Keidar, I., Kliot, G., Shraer, A.: Brahms: Byzantine Resilient Random Membership Sampling. Computer Networks **53** (2009) 2340–2359 A former version appeared in the 27th ACM Symposium on Principles of Distributed Computing (PODC), 2008.
9. Awerbuch, B., Scheideler, C.: Group Spreading: A Protocol for Provably Secure Distributed Name Service. In: Proceedings of the 31rst International Colloquium on Automata, Languages and Programming (ICALP). (2004)
10. Jesi, G.P., Montresor, A., van Steen, M.: Secure Peer Sampling. Computer Networks **54**(12) (2010) 2086–2098
11. Singh, A., Ngan, T.W., Druschel, P., Wallach, D.S.: Eclipse Attacks on Overlay Networks: Threats and Defenses. In: Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM). (2006)
12. Douceur, J., Donath, J.S.: The Sybil Attack. In: Proceedings of the 1rst International Workshop on Peer-to-Peer Systems (IPTPS). (2002)
13. Awerbuch, B., Scheideler, C.: Towards a Scalable and Robust Overlay Network. In: Proceedings of the 6th International Workshop on Peer-to-Peer Systems (IPTPS). (2007)
14. Anceaume, E., Ludinard, R., Sericola, B.: Performance evaluation of large-scale dynamic systems. SIGMETRICS Performance Evaluation Review **39**(4) (2012) 108–117
15. Anceaume, E., Brasileiro, F., Ludinard, R., Sericola, B., Tronel, F.: Dependability Evaluation of Cluster-based Distributed Systems. International Journal of Foundations of Computer Science (IJFCS) **5**(22) (2011)
16. Jelasity, M., Voulgaris, S., Guerraoui, R., Kermarrec, A.M., van Steen, M.: Gossip-based Peer Sampling. ACM Transaction on Computer System **25**(3) (2007)
17. Karp, R., Schindelhauer, C., Shenker, S., Vocking, B.: Randomized Rumor Spreading. In: the 41st Annual Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society (2000) 565
18. Voulgaris, S., Gavidia, D., van Steen, M.: CYCLON: Inexpensive Membership Management for Unstructured P2P Overlays. Journal of Network System Management **13**(2) (2005) 197–217
19. Zhong, M., Shen, K., Seiferas, J.: Non-uniform Random Membership Management in Peer-to-Peer Networks. In: Proceedings of the 24th Annual Joint Conference of the Computer and Communications Societies (INFOCOM), IEEE Press (2005)
20. Awan, A., Ferreira, R.A., Jagannathan, S., Grama, A.: Distributed Uniform Sampling in Unstructured Peer-to-Peer Networks. In: Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS). (2006)
21. Castro, M., Druschel, P., Ganesh, A., Rowstron, A., Wallach, D.S.: Secure Routing for Structured Peer-to-peer Overlay Networks. In: Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI), ACM (2002)
22. Ratnasamy, S., Francis, P., Handley, M., Karp, R.M., Shenker, S.: A scalable content-addressable network. In: Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM). (2001)
23. Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M.F., Dabek, F., Balakrishnan, H.: Chord: a Scalable Peer-to-Peer Lookup Protocol for Internet Applications. IEEE/ACM Transaction on Networks **11**(1) (2003) 17–32
24. Rowstron, A.I.T., Druschel, P.: Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middlware). (2001)

25. Hildrum, K., Kubiatowicz, J.: Asymptotically Efficient Approaches to Fault-tolerance in Peer-to-Peer Networks. In: Proceedings of the International Symposium on Distributed Computing (DISC). (2003)
26. Fiat, A., Saia, J., Young, M.: Making Chord Robust to Byzantine Attacks. In: Proceedings of the Annual European Symposium on Algorithms. (2005) 803–814
27. Anceaume, E., Brasileiro, F., Ludinard, R., Ravoaja, A.: PeerCube: an Hypercube-based P2P Overlay Robust against Collusion and Churn. In: Proceedings of the IEEE International Conference on Self-Adaptive and Self-Organizing Systems. (2008)
28. Condie, T., Kacholia, V., Sank, S., Hellerstein, J.M., Maniatis, P.: Induced Churn as Shelter from Routing-Table Poisoning. In: Proceedings of the International Network and Distributed System Security Symposium (NDSS). (2006)
29. Liu, D., Ning, P., Du, W.: Detecting Malicious Beacon Nodes for Secure Location Discovery in Wireless Sensor Networks. In: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS). (2005)
30. Anceaume, E., Busnel, Y., Gambs, S.: Uniform and Ergodic Sampling in Unstructured Peer-to-Peer Systems with Malicious Nodes. In: Proceedings of the 14th International Conference On Principles Of Distributed Systems (OPODIS). Volume 6490. (2010)
31. Anceaume, E., Busnel, Y., Gambs, S.: Characterizing the adversarial power in uniform and ergodic node sampling. In: Proceedings of the 1st International Workshop on Algorithms and Models for Distributed Event Processing (AlMoDEP), ACM (2011)
32. Busnel, Y., Beraldi, R., Baldoni, R.: On the Uniformity of Peer Sampling based on View Shuffling. Elsevier Journal of Parallel and Distributed Computing **71**(8) (2011) 1165–1176
33. Gurevich, M., Keidar, I.: Correctness of Gossip-Based Membership under Message Loss. In: Proceedings of the 28th annual Symposium on Principles of distributed computing (PODC), Calgary, AL, Canada, ACM Press (2009)
34. Yu, H., Gibbons, P.B., Kaminsky, M., Xiao, F.: SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks. In: Proceedings of the IEEE Symposium on Security and Privacy (SP). (2008)
35. Yu, H., Kaminsky, M., Gibbons, P.B., Flaxman, A.: SybilGuard: Defending against Sybil Attacks via Social Networks. In: Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM). (2006)
36. Godfrey, P.B., Shenker, S., Stoica, I.: Minimizing churn in distributed systems. In: Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM). (2006)
37. Cover, T., Thomas, J.: Elements of information theory. Wiley New York (1991)
38. Ali, S.M., Silvey, S.D.: General Class of Coefficients of Divergence of One Distribution from Another. Journal of the Royal Statistical Society. Series B (Methodological) **28**(1) (1966) 131–142
39. Anceaume, E., Busnel, Y.: A distributed information divergence estimation over data streams. IEEE Transactions on Parallel and Distributed Systems **99**(PrePrints) (2013)
40. Monge, G.: Mémoire sur la théorie des déblais et des remblais. Histoire de l'Academie royale des sciences, avec les Mémoires de Mathématique et de Physique (1781) 666–704