Scalable Approximate Query Tracking over Highly Distributed Data Streams *

Nikos Giatrakos
Technical University of Crete
ngiatrakos@softnet.tuc.gr

Antonios Deligiannakis
Technical University of Crete
adeli@softnet.tuc.gr

Minos Garofalakis
Technical University of Crete
minos@softnet.tuc.gr

ABSTRACT

The recently-proposed Geometric Monitoring (GM) method has provided a general tool for the distributed monitoring of arbitrary non-linear queries over streaming data observed by a collection of remote sites, with numerous practical applications. Unfortunately, GM-based techniques can suffer from serious scalability issues with increasing numbers of remote sites. In this paper, we propose novel techniques that effectively tackle the aforementioned scalability problems by exploiting a carefully designed sample of the remote sites for efficient approximate query tracking. Our novel sampling-based scheme utilizes a sample of cardinality proportional to $\sqrt{N}$ (compared to N for the original GM), where N is the number of sites in the network, to perform the monitoring process. Our experimental evaluation over a variety of real-life data streams demonstrates that our sampling-based techniques can significantly reduce the communication cost during distributed monitoring with controllable, predefined accuracy guarantees.

1. INTRODUCTION

Efficient data stream processing algorithms have become an integral part of real-time monitoring applications, from network traffic monitoring to financial or stock data analysis and sensor data querying. Streaming tuples are rapidly produced in a number of geographically dispersed sites (routers, ATMs, sensor nodes etc) and are *continuously* processed online to provide continuous up-to-date query answers destined to support decision making procedures such as DDoS attacks, fraudulent transactions, market trend predictions, and tsunami wave detection, in a timely manner. In such distributed settings, it is imperative to design efficient algorithms that reduce the communication burden during the continuous monitoring process [4, 7], since either the available bandwidth is limited, or data transmission is a crucial factor that reduces network lifetime (e.g., for battery-powered sensor nodes [28]).

The problem of efficiently tracking the value of a function (often compared to some predefined threshold) over the union of local streams in a large-scale distributed system, lies at the core of several recent research efforts [9, 35, 7, 8, 10, 38]. Monitoring tasks may involve functions that are simple linear aggregates, such as checking whether the sum of a distributed set of variables exceeds a predetermined threshold [12, 38], thresholded counts of items [23] or frequently occurring items in a set of distributed streams [29]. More complicated function monitoring may involve holistic aggregates [9, 38], self-join as well as stream-join operations [8, 15], or general, *non-linear* function tracking [35].

The original work of Sharfman et al. [35] is the first to propose a generic, Geometric Monitoring (GM) method for monitoring *any non-linear function* f over the global average of vectors maintained at the distributed sites, with respect to some threshold T, i.e., monitoring whether $f(\cdot) \lessgtr T$. The GM method is a very powerful technique that has already been exploited in a wide range of applications, including: (i) outlier detection in sensor networks [1], where the monitored function is any of the L_p norms, cosine similarity, Extended Jaccard Coefficient, or correlation coefficient; (ii) tracking range, norm-aggregate and join-aggregate queries [15, 25] over distributed data streams; (iii) monitoring fragmented skyline queries [30]; (iv) detecting machines that are about to become faulty in data centers [13]; and, (v) distributed online prediction [22] by dynamically monitoring the accuracy of distributed local models. In a nutshell, the GM method can offer a general solution to any useful monitoring task (expressed over the data collected by distributed data sources), in which continuous data communication to a central site is not feasible, due to either bandwidth or energy constraints.

The basic concept utilized in [35], in order to monitor general functions, is the identification and distributed monitoring of a subset of the input domain of f where this average vector may lie (rather than the function value itself). Each site maintains a local measurements vector, produced based on the data of the site and the monitored function. The local measurements vector helps determine a monitoring zone, constructed as a proper hypersphere, of the site. Each site then checks if the function at any point of its monitoring zone exceeds/falls below the threshold T, in which case it raises an alarm that is resolved by additional communication. Geometrically, the area of the input domain that needs to be monitored is equivalent to the convex hull of the sites' local measurements vectors, which is guaranteed to contain the average vector. The union of these hyperspheres is proven to cover this convex hull, which ensures that each point of the convex hull is monitored by at least one site. Figure 1 schematically presents the above concept, details will follow in Section 3. In terms of communication efficiency, central data collection is constantly postponed until at least one site finds that input points inside its local sphere may have caused the function to cross T. In the latter case, local vectors are

*This work was supported by the European Commission under ICT-FP7-FERARI-619491 (Flexible Event pRocessing for big dAta aRchItectures).

SIGMOD'16, June 26-July 01, 2016, San Francisco, CA, USA

DOI: http://dx.doi.org/10.1145/2882903.2915225

collected in a central source, an up-to-date global average is computed and f is checked to assess whether it truly crossed T or not.

In this work, we demonstrate that, despite the generic nature of the GM method as a distributed tracking scheme, it faces significant scalability issues as the number of (distributed) remote sites increases. First, we show that increasing the network scale (i.e., the number of sites, N) increases the size of the monitored area (union of balls), thus resulting in an excessive number of central data collections, even when the tracked function has not actually crossed the threshold. Such unnecessary centralization choices are referred to as False Positives (FPs). Second, the communication cost of central data collections increases proportionally to N and, thus, the total bandwidth consumption throughout the tracking process increases significantly for large values of N. Third, the above problems become even more pronounced when the monitoring query is defined over the sum (rather than the average) of the local measurements vector at the sites. This is often the case in practice, e.g., when tracking relational join aggregates over frequency distribution vectors fragmented across the sites [15].

Having identified the degree of distribution (N) as a key limitation of the GM method, we introduce an algorithmic framework that exploits a small sample (proportional to only $O(\sqrt{N})$) of the sites to perform the monitoring process, and we formally study the properties of our sampling-based geometric scheme and how it addresses the scalability issues of GM. In a nutshell, our scalable approximate monitoring algorithm exploits *Horvitz-Thompson sampling estimators* [5, 34] over a carefully built sample of the sites in order to construct low-error approximations of the average vector; furthermore, it employs multi-dimensional tail-probability bounds and thorough geometric analysis to control the effect of these approximations on the accuracy of GM. Our approach can considerably decrease the amount of false positive data centralizations and the communication burden on the network at the cost of potentially causing a few False Negatives (FNs), i.e., missing a true threshold crossing of the monitored function. Note that if such missed *threshold violations* are quickly corrected (by appropriate detection) in the immediate future, and controlled in number, then they are generally acceptable in monitoring applications where GM is employed. For example, detecting machines likely to fail in the future in large data centers (in [13]), reporting when the size of a self-join query exceeds a given threshold (in [15, 25]), or detecting when distributed online learning models have become inaccurate (in [22]) with a slight delay (e.g., over the next few data collections) has little impact on the importance and gains of distributed GM function tracking, especially when the rate of such FNs is low and can be explicitly controlled.

We provide an extensive theoretical analysis exhibiting that the amount of such false negatives is controllable, given predefined accuracy guarantees based on the desired level of communication efficiency (which is directly linked to the site-sample size). Moreover, in our experiments (Section 6.4, also see Appendix) we also demonstrate that even if FNs occur (in a controlled manner), the missed threshold crossings are detected soon afterwards, most often in the next centralization decision. We re-iterate that the communication efficiency of our techniques is essential not only in applications collecting massive amounts of data (e.g., in IP-network monitoring [11]), but also in applications characterized by power and bandwidth restrictions, as in battery-powered wireless sensor nodes, where communication is the key determinant of sensor battery life [28]. Our contributions are summarized as follows:

- We point out the limitations of the GM approach in highly distributed environments, that result in excessive false positive data centralization decisions and prohibitive communication cost.

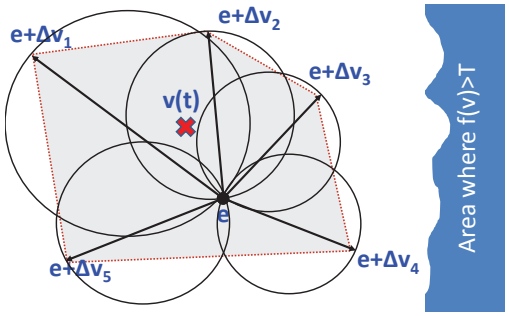


Figure 1: Illustration of GM at a given time point t. The monitored convex hull is depicted in gray, while the position of e and the current $v(t)$ are shown as well. Black spheres refer to the local constraints constructed by sites. Since none of them crosses the threshold surface, no synchronization is needed.

- Targeting the root of the above limitations, we introduce approximate algorithms for tracking any non-linear function that use only a carefully built sample of the network sites. Our techniques provide the ability to tune the sample size, and thus the anticipated communication cost according to application-defined approximation accuracy requirements. We provide an extensive theoretical analysis to formally quantify the accuracy vs. communication cost trade-off of our sampling-based techniques.
- We develop a methodology according to which a proper site sampling function is formed and parameterized. Our proposed sampling function yields a sample size proportional to only $O(\sqrt{N})$ of the total sites in the network, in contrast to $O(N)$ sites in the original GM method and its variants [16, 17].
- We conduct extensive experimentation on two real datasets, using a variety of different functions, threshold values and network sizes. Our performance comparisons are against not only the vanilla GM approach [35], but also against the balancing optimization proposed in [35] and the recently proposed prediction-based GM [16, 17]. Although these optimizations are orthogonal to our proposed framework, we show that in highly distributed environments our algorithms can ensure in most cases *one and up to two orders of magnitude less bandwidth consumption* compared to other methods, while also significantly reducing the *per site* cost (transmitted messages), even without exploiting any of these optimizations within our sampling-based scheme.

2. RELATED WORK

Abundant works have focused on efficiently performing monitoring tasks in distributed data streams. Some of them were already mentioned in Section 1, while [4] presents a recent survey on related techniques. Here, we concentrate on studies closely related to the GM framework and site sampling techniques.

Monitoring General Threshold Functions. The basic operation of the GM framework was introduced in [35]. Our work, after pointing out the shortcomings that arise in highly distributed data streams, proposes techniques to effectively confront existing scalability issues of the GM framework. [35] also proposes a balancing optimization to the basic scheme to further reduce the communication cost of their approach. However, the aforementioned balancing technique is merely a heuristic for which we experimentally (Section 6) show that is hardly adequate in highly distributed settings.

The GM framework has been enhanced in [36], where ellipsoidal instead of spherical local constraints are considered. These

methods are orthogonal to the algorithms that we develop. However, [36] assumes that data follows a multivariate normal distribution; furthermore, [36] also suffers from scalability issues, since using ellipsoids instead of hyperspheres neither alters the fact that the higher N is, the higher the amount of the monitored area nor reduces the cost of a false positive central data collection.

The recently introduced prediction-based GM [16, 17] constitutes another technique that is orthogonal to the methods that we present in our work. However, [16, 17] heavily depends on accurate predictions of the local vectors maintained at each site. Nonetheless, accurately predicting several vector components over many sites becomes increasingly harder with the increase of the network scale. This is also demonstrated in our experimental evaluation.

A number of works design techniques that are geometric in nature but, contrary to our approach, focus only on specific types of functions. [32] considers functions with bounded deviation and introduces a tentative bound algorithm to monitor threshold queries in distributed databases (rather than distributed streams), while [33] focuses on vectorial top-k aggregation queries over distributed databases. Moreover, the work in [15] couples sketch summaries with the GM framework focusing on join aggregates, special cases of L_2-norms and range aggregates (e.g., quantiles, wavelets, and heavy-hitters over the streams). The work in [1] utilizes GM for outlier detection in sensor networks, reducing the problem to multiple monitoring tasks, with each task involving only the pair of nodes whose similarity is to be monitored. [24] proposes an approach, for monitoring *heterogeneous* streams by defining constraints tailored to fit the specific data distributions of sites.

Site Sampling Techniques. The sampling component that we develop as part of our tracking schemes allows the continuous monitoring on any generic function $f : \mathbb{R}^d \rightarrow \mathbb{R}$. This is the main breakthrough that distinguishes our contributions compared to individual site sampling techniques that can only handle linear functions such as counts and frequencies [39, 19, 20] or second frequency moments [27]. Besides this crucial distinction, our sampling component possesses more generic characteristics compared to existing site sampling approaches [39, 19, 20, 27]. These characteristics can be summarized as follows: (a) our analysis, from approximation quality issues to extracted estimators and expected communication savings is *multidimensional* in nature. On the contrary, [39, 19, 20, 27] define sampling schemes operating on a single dimension, (b) our techniques are destined to support *continuous monitoring* procedures while [39, 19] focus on one-shot queries, (c) Our algorithms do not incorporate any assumption about local input monotonicity or boundedness and are capable of handling *unbounded, non-monotonic local inputs* (updates). The techniques in [39, 19, 20, 27] assume bounded updates, while [39, 19, 20] are restricted to positive inputs only. (d) [39, 19, 20, 27] are focused on ensuring *accuracy relative* to the current global frequency or count which can be known only after acquiring the sample. Our algorithms abide by predetermined accuracy constraints, which is a necessary feature in our setting. Due to the above limitations, [39, 19, 20, 27] are not applicable in our setup.

3. PRELIMINARIES & MOTIVATION

In Section 3.1 we provide an overview of the Geometric Monitoring (GM) framework. Section 3.2 motivates the need for our approach, as it provides intuition on why the existing GM framework may lead to increased communication overhead when either the number of network sites increases, or when the monitored function is parameterized with the sum, rather than the global average, of local measurements vectors.

3.1 Geometric Monitoring Basics

As in previous works [9, 35, 6, 8, 36], we assume a distributed, two-tiered setting, where data arrives continuously at N geographically dispersed sites. At the top tier, a central coordinator exists that is capable of communicating with every site, while pairwise site communication is only allowed via the coordinating source. Each site S_i, $i \in [1..N]$ participating at the bottom tier periodically receives updates on its local stream and maintains a d-dimensional *local measurements vector* $v_i(t)$, capturing the current state of its local stream. The *global measurements vector (i.e., stream)* $v(t)$ at any given timestamp t, is computed as the average of $v_i(t)$ vectors, $v(t) = \frac{\sum_{i=1}^{N} v_i(t)}{N}$. The coordinator aims to continuously monitor if the value of a function $f(v(t))$, parameterized by the global average $v(t)$, lies above/below a given threshold T. We term the area of the input domain where $f(v(t)) = T$ as the *threshold surface*.

Assume that at a previous time instant t_s, the coordinator has collected the local $v_i(t_s)$ vectors. Using $e(t_s)$ to distinguish the global average $v(t_s)$ at t_s, the coordinator computes $e(t_s) = \frac{\sum_{i=1}^{N} v_i(t_s)}{N}$ at that time, subsequently broadcasting $e(t_s)$ to the sites in the bottom tier. The previous process is referred to as a *synchronization* step. Note that, until the next synchronization, the coordinator's view of the global vector stays constant at $e(t) = e(t_s)$. Following [35], upon receiving $e(t)$, sites keep up receiving updates of their local streams and accordingly maintain their $v_i(t)$ vectors. At any given timestamp, each site S_i individually computes a *deviation vector* $\Delta v_i(t) = v_i(t) - v_i(t_s)$, which depicts the change that the local vector has undergone since t_s. By attaching the deviation vector to $e(t)$, sites compute their *drift vectors* as $e(t) + \Delta v_i(t)$. Since $v(t) = \frac{\sum_{i=1}^{N} v_i(t)}{N} = e(t) + \frac{\sum_{i=1}^{N} (v_i(t) - v_i(t_s))}{N} = \frac{\sum_{i=1}^{N} (e(t) + \Delta v_i(t))}{N}$, $v(t)$ constitutes a convex combination of the drift vectors.

Consequently, $v(t)$ will always lie in the convex hull formed by the $\Delta v_i(t)$ vectors translated by $e(t)$, as depicted in Figure 1 for $d = 2$, $N = 5$: $v(t) \in Conv\ (e(t) + \Delta v_1(t),\ \ldots,\ e(t) + \Delta v_N(t))$. If the convex hull does not intersect the *inadmissible* area of the input domain (on the right of Figure 1), where the monitored inequality is reversed (from $f(v(t)) > T$ to $f(v(t)) < T$ or vice versa), it is assured that $v(t)$ cannot lie in that area either. Hence, our monitoring problem is transformed to how to decide in a distributed manner whether the convex hull intersects the threshold surface.

It has been proven [35] that if sites locally construct hyperspheres $B(e(t) + \frac{1}{2}\Delta v_i(t), \frac{1}{2}\|\Delta v_i(t)\|)$, centered at $e(t) + \frac{1}{2}\Delta v_i(t)$ with radius $\frac{1}{2}\|\Delta v_i(t)\|$, then:

$$Conv(e(t) + \Delta v_1(t), \ldots, e(t) + \Delta v_N(t)) \subset \bigcup_{i=1}^{N} B(e(t) + \tfrac{1}{2}\Delta v_i(t), \tfrac{1}{2}\|\Delta v_i(t)\|)$$

That is, the union of these hyperspheres is always guaranteed to cover the convex hull of the translated local drifts (in any dimensionality). Thus, having constructed $B(e(t) + \frac{1}{2}\Delta v_i(t), \frac{1}{2}\|\Delta v_i(t)\|)$, each site individually checks for an intersection of its local sphere with the threshold surface. In case an intersection exists in at least one S_i, a *local violation* occurs at S_i indicating that the convex hull and, thus, $v(t)$ **may** have crossed the threshold surface. Hence, a synchronization takes place where the coordinator collects the $v_i(t)$ vectors and assesses whether $f(v(t))$ truly switched sides ($\lessgtr$) with T. It then computes the new $e(t)$ and communicates it back to the sites. From this point forward, the tracking process can proceed as described above. If no local violation occurs (as in the example of Figure 1), then no communication is necessary.

EXAMPLE 1. *Consider a number of sensors in a server room monitoring in real-time whether uniform relative Humidity (rH) conditions exist for all the machines to operate normally. Too dry*

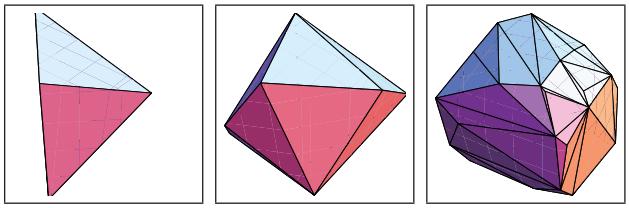


Figure 2: The effect of network scale on the monitored area ($d=3$). All Δv_i vectors are randomly chosen from the unit cube, the front view of which is the box included in each figure. As the network scale increases, more Δv_i vectors grow on the $d-$dimensional plane. Inevitably, the convex hull that needs to be monitored is enlarged favoring FP decisions.

Table 1: Frequently Used Symbols

Symbol	Description
N	The number of sites of the bottom tier
d	Dimensionality of the input domain
S_i	The i-th site, $S_i \in \{S_1,\ldots,S_N\}$
t_s	Time-point of the last synchronization
$v_i(t)$	Local measurements vector at S_i at time t
$\Delta v_i(t)$	Deviation vector at S_i at time t (equals $v_i(t)-v_i(t_s)$)
$v(t),\hat{v}(t)$	Global average & its statistic estimator at time t
$e(t)=v(t_s)$	Global average vector computed after a synchronization
$B(c,\rho)$	Hypersphere centered at c with ρ sized radius
(ε,δ)	Approximation pair denoting that $\hat{v}(t)$ should lie within ε distance from $v(t)$ with probability as least $1-\delta$
g_i	Sampling function $0 \leq g_i \leq 1$ for site S_i
ε_T	Minimum distance of $e(t)$ from the threshold surface

air will result in the build up of static electricity on the systems, while high humidity slowly damages equipment. To ensure uniform rH, the application continuously checks $Var(v(t)) > T$, where Var denotes the variance function [13]. Each sensor maintains a periodically updated local vector $v_i(t) = [c_i^2(t), c_i(t)]$, with $c_i(t)$ being the recent rH measurement at S_i. Variance is computed by $Var(v(t)) = \frac{1}{N}\sum_{i=1}^{N} c_i^2 + (\frac{1}{N}\sum_{i=1}^{N} c_i)^2$ and thus $v(t) = \left[\frac{1}{N}\sum_{i=1}^{N} c_i^2, \frac{1}{N}\sum_{i=1}^{N} c_i\right]$. If $v_i(t_s)$ is the last rH value S_i communicated to the coordinator, $\Delta v_i(t) = [c_i^2(t) - c_i^2(t_s),\ c_i(t) - c_i(t_s)]$. Given these, local spheres are constructed and potential threshold crossings are assessed.

Communication savings are ensured by postponing a synchronization until some site finds its local sphere intersecting the threshold surface. Also note that the convex hull or its superset, the union of local spheres, may cross the threshold surface, while the actual position of $v(t)$ may not be in the intersecting part. As a result, the framework may cause synchronizations when $f(v(t))$ has not crossed the threshold (false positives). Table 1 summarizes the main notation used in this paper.

3.2 Existing Scalability Issues

We now explain why the GM framework may result in increased communication in either highly distributed networks, or in cases where the monitored function is parameterized with the sum of the local measurements vectors.

High N values $\Rightarrow$ proneness to FP synchronizations. As already described, the area that GM needs to track is the convex hull $Conv(e+\Delta v_1(t),\ \ldots,\ e+\Delta v_N(t))$. It is not difficult to see that the more sites participate in the distributed monitoring process (and, thus, contribute their $e+\Delta v_i(t)$ in the formation of this convex hull), the larger the tracked area will be. Figure 2 schematically exhibits the effect of progressively increasing the network scale from $N=5$ sites to $N=10$ and $N=100$ in a 3-d space by randomly picking additional Δv_is from the unit cube. Larger values of N yield a convex hull that tends to cover the entire unit cube (boxed area in Fig. 2). Moreover, the hyperspheres maintained by each site in order to include the expanded convex hull will cover an even larger, compared to the expanded convex hull, area of the input domain because $Conv(e(t)+\Delta v_1(t),\ \ldots,\ e(t)+\Delta v_N(t))$ is a subset of $\bigcup_{i=1}^{N} B(e(t)+\frac{1}{2}\Delta v_i(t),\ \frac{1}{2}\|\Delta v_i(t)\|)$. This raises the potential for a larger number of FP synchronization decisions as the number of sites increases.

Note that the cost of a FP synchronization decision is equivalent to $N+1$ messages, assuming the coordinator is equipped with broadcast capabilities, or $2N$ otherwise, and that all sites are required to participate in this process. This not only increases the total communication cost per FP as the number of sites N increases, but also increases the cost per site, since a site transmits messages each time at least one site exhibits a local violation.

Problematic application on monitoring sum-parameterized functions. Consider the case where the function that needs to be monitored is parameterized with the sum rather than the global average, i.e., $f(v_{sum}(t)) \lessgtr T$ with $v_{sum}(t) = N \cdot v(t) = \sum_{i=1}^{N} v_i(t)$. Such functions, for instance, include L_2 norms during approximate function monitoring queries [15, 17], or statistics like the Mutual Information function in news monitoring applications [16].

Let this time $e_{sum}(t) = v_{sum}(t_s)$. The above equation shows that $v_{sum}(t)$ can be expressed as a convex combination of $(e(t)+\Delta v_i(t))$ vectors scaled by N since $N\cdot(e(t)+\Delta v_i(t)) = e_{sum}(t)+N\cdot\Delta v_i(t)$. As a consequence, $v_{sum}(t)$ lies inside the scaled convex hull $Conv(N\cdot(e(t)+\Delta v_1(t)),\ldots,N\cdot(e(t)+\Delta v_N(t)))$.

We can now perform the monitoring in a slightly modified way from the original GM scheme of Section 3.1. The difference is that the new local constraint in each site S_i is scaled by N, i.e., $B(N\cdot e(t)+\frac{N}{2}\Delta v_i(t),\ \frac{N}{2}\|\Delta v_i(t)\|)$ which is checked so as to assess whether it invades the threshold surface. Consequently, in highly distributed settings, the degree of distribution N has once again an undesirable impact on the size of the local constraints, increasing the proneness of the framework to FP synchronizations.

4. SCALABLE QUERY TRACKING

Having shown that the degree of distribution N is the factor that renders GM practically inefficient, we now design a sampling-based framework that overcomes these scalability issues. Section 4.1 first formally presents a set of requirements that any candidate sampling-based scheme for GM monitoring should adhere to, in order to both guarantee efficiency in terms of bandwidth consumption and compliance with application-defined accuracy requirements.

Then, in Section 4.2, we present our generic sampling-based GM scheme in detail. The intuition behind our approach is that, instead of monitoring the entire convex hull formed by the N sites, we choose to track a narrower convex hull composed of a carefully-crafted random sample of sites. The latter area constructs a subset of $Conv(\{e+\Delta v_i\} : \forall S_i \in \{S_1,\ldots,S_N\})$ reducing the tracked space

and warding off FP synchronizations. For ease of exposition, we henceforth omit the temporal reference t where appropriate.

4.1 Efficiency and Accuracy Requirements

In addition to choosing fewer sites to reduce the monitored area, our sampling-based geometric scheme should be able to *guarantee improved communication efficiency* by ensuring that its inscribed local constraints are fully contained within the local constraints $\bigcup_{i=1}^{N} B(e+\frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$ of the original GM method (Section 3.1). This guarantees that the area tracked by the sampled sites does not cross the threshold surface before the union of balls of the conventional GM does and, thus, additional FP synchronizations **cannot** be caused.

REQUIREMENT 1. **[Efficiency]** *Let Sur_{sample} denote the area monitored by a candidate sampling-based monitoring scheme. To guarantee efficiency by reducing FP synchronizations, inclusion $Sur_{sample} \subseteq \bigcup_{i=1}^{N} B(e+\frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$ should hold.*

Since the monitoring process utilizes a small sample (subset) of the sites available in the network, it will be approximate in nature. At any given time t, our sampling-based geometric techniques monitor an *unbiased* (i.e., $E(\hat{v}(t)) = v(t)$) estimation $\hat{v}(t)$ of the true global average $v(t)$ originating from only a sample $K \subseteq \{S_1, \ldots S_N\}$ of sites' local vectors. We wish to keep the estimation error controllable and tunable based on *a priori* defined accuracy requirements. To control the approximation error, we employ an (ε, δ) approximation scheme. More precisely, for a priori given $0 < \delta \leq 1$, $\varepsilon > 0$ we shall require $v \in B(\hat{v}, \varepsilon)$ with high probability, at least $1-\delta$.

REQUIREMENT 2. **[Approximation Quality]** *At any given time, the estimation $\hat{v}$ monitored by the sampled sites should not exceed $\varepsilon-$ distance from the true v, with high probability $1-\delta$; that is, for given $0 < \delta \leq 1$, $\varepsilon > 0$: $P(v \notin B(\hat{v}, \varepsilon)) \leq \delta$.*

Due to the fact that $v(t)$ is monitored in an approximate way tuned according to (ε, δ), it is possible that a synchronization is prevented while $v(t)$ truly switched side with respect to the threshold surface. Our sampling-based scheme should enable applications to explicitly tune the probability P_{FN} of such False Negative (FN) events. This requires an additional (application-defined) input parameter apart from (ε, δ).

Nonetheless, specifying a triplet of parameters two of which refer to the input domain rather than the function value may be confusing from an application viewpoint. Application accuracy requirements should be expressed in a simple way that abstracts the actual details of the input domain of the tracking process. While it is more natural for the end user to directly specify P_{FN}, for ease of presentation we assume that the parameter specified is the δ parameter of Requirement 2. We demonstrate in Section 5 that P_{FN} is directly linked to δ. Then, our scheme can accordingly tune not only the probability of a FN, i.e., P_{FN}, but also the approximation quality in the (ε, δ) scheme. To do so, ε should be expressed as a function of δ, i.e., $\varepsilon = \varepsilon(\delta)$.

REQUIREMENT 3. **[Tunable Accuracy]** *At any given time, the proposed sampling-based monitoring algorithm should possess the ability to receive a sole tolerance value $0 < \delta \leq 1$ and self-tune its Approximation Quality i.e., (ε, δ) and FN rate i.e., P_{FN}.*

Hence, we assume that the application expresses its monitoring needs in the form: $f(v(t)) \gtrless T$, $\delta : 0 < \delta \leq 1$. We then proceed in describing the generic operation of our sampling-based framework.

4.2 Our Generic Sampling-Based Scheme

We now present our sampling-based GM algorithm and demonstrate how it satisfies Requirements 1-3. A key idea in our scheme is to independently sample each site S_i with a different probability g_i that depends on various factors. Our discussion in this section assumes that these sampling probabilities g_i have been determined, deferring the details and analysis of the g_i computation to Section 5.

Algorithmic Sketch. Consider that at a previous time, a synchronization has taken place, as described in Section 3.1, and that e has been transmitted to all $S_i \in \{S_1, \ldots, S_N\}$. At any subsequent timestamp, each S_i keeps receiving updates of its local vector v_i and computes Δv_i. In our sampling scheme, S_i constructs a local constraint in the form of a hypersphere only if it finds itself included in the sample K of sites participating in the monitoring process, i.e., $S_i \in K$. In order to determine if $S_i \in K$, each site independently flips a biased coin with success probability of g_i, where $g_i \in [0,1]$ is a sampling function independently computed by each site. Each $S_i \in K$ inscribes a sphere $B(e+\frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$ which is checked for threshold crossing. In case at least one $S_i \in K$ detects a threshold crossing of its local constraint, it calls for a synchronization. During the synchronization, the coordinator initially broadcasts a message requiring only the sampled sites to contribute their Δv_i vectors. Using these vectors, it derives an unbiased estimate $\hat{v}$ (using Estimator 1 below) of v and checks (Requirement 2) whether $B(\hat{v}, \varepsilon)$ crosses T. If it does not, then the coordinator deduces that this was a FP alarm with probability $1-\delta$ and the tracking continues unaffected. Otherwise, a full synchronization takes place, where also all $S_i \notin K$ contribute their Δv_is for a new e vector to be computed and communicated to the underlying sites. In the latter case, the coordinator believes that a true threshold violation may have taken place and probes the whole set of sites to compute the exact value of e. This is required to avoid an additive error in the approximation of $\hat{v}$ (see Estimator 1) as the tracking process continues.

We now provide the details of our monitoring scheme, explain our design choices (as described in the above algorithmic sketch), and discuss the accuracy guarantees of our technique. In each of these steps, we examine the satisfiability of Requirements 1-3 in conjunction with our algorithmic sketch. For ease of exposition, we start our discussion with Requirement 2.

Monitored Estimator and Approximation Quality Requirement. Consider a multivariate random variable $\Delta' v_i = \frac{\Delta v_i}{g_i}$ with probability g_i, and zero otherwise. Notice that $E[\Delta' v_i]$ is a $d-$dimensional vector and $E[\Delta' v_i] = [E[\Delta' v_{i1}], \cdots, E[\Delta' v_{id}]]$, where $\Delta' v_{ij}$ denotes the j-th component (dimension) of the vector $\Delta' v_i$. We demonstrate in Lemma 1 that, based on the drift vectors $e + \frac{\Delta v_i}{g_i}$ of the set K, an unbiased estimate $\hat{v}$ of the global average v can be derived at any given time stamp t utilizing a *Horvitz-Thompson Estimator* [5, 34]:

$$\hat{v} = e + \frac{\sum_{i=1}^{N} \Delta' v_i}{N} = e + \frac{\sum_{S_i \in K} \frac{\Delta v_i}{g_i}}{N} \quad (1)$$

Note that the global average is $v = e + \Delta v$, with $\Delta v = \sum_{i=1}^{N} \Delta v_i / N$. Hence, Estimator 1 estimates Δv as $\hat{\Delta v} = \sum_{S_i \in K} \frac{\Delta v_i}{g_i} / N$. The estimator weighs each sampled site with $1/g_i$. The reason for this is fairly intuitive: If site S_i, which is sampled with probability g_i, individually appears in the sample, then, on average, we expect to have $1/g_i$ sites with similar probabilities in the full population (since $g_i \cdot 1/g_i = 1$); thus, the single occurrence of S_i in the sample is essentially a "representative" of $1/g_i$ sites in the full population [21, 34].

LEMMA 1. *For Estimator 1 the following hold:*
(a) Estimator 1 is an unbiased estimator of v when sampling $\forall S_i \in \{S_1, \ldots, S_N\}$ *with* $0 \leq g_i \leq 1$.
(b) $E[\hat{v}] \in Conv(\ e + \Delta v_1, \ldots, e + \Delta v_N)$
(c) $\hat{v} \in Conv(\{e + \frac{\Delta v_i}{g_i}\} : \forall S_i \in K)$

PROOF. See Appendix □

Since Estimator 1 is unbiased, we can utilize standard tail inequalities [14] to satisfy Requirement 2. Note that we do not assume independence of individual dimensions of either local, or global vectors that we examine. Nonetheless, according to our algorithmic sketch, S_is independently decide to include themselves in K or not, based on g_i. The Vector Bernstein's Inequality [2] (presented below) will be particularly useful in our subsequent analysis.

Vector Bernstein's Inequality [2]. Let $y_1, \ldots, y_N$ be independent random vectors with $E[y_i] = 0$. Let $B > 0$ denote an upper bound on $\|y_i\|$ (i.e., $\|y_i\| \leq B$), and let $\sigma^2 \geq \sum_{i=1}^{N} E[\|y_i\|^2]$. Then, for all $0 < \delta \leq 1$ and $0 \leq \varepsilon \leq \sigma^2/B$ such that $\varepsilon = (1 + \sqrt{\ell n(1/\delta)}) \cdot \sigma$:

$$P(\| \sum_{i=1}^{N} y_i \| \geq \varepsilon) \leq \delta \quad \square \tag{2}$$

The inequality states that if we add N random vectors of bounded length which move around zero, their sum will produce a vector placed near (no farther than ε) to zero with probability at least $1 - \delta$. The proximity (ε) of the vector sum to zero depends on an upper bound σ on the overall standard deviation [1] and the chosen probability bound δ. Note that the above bound does not depend on the dimensionality d of the vectors. In our case, each y_i corresponds to $\frac{\Delta' v_i - \Delta v_i}{N}$. Moreover, $B \geq \{\| \frac{\Delta v_i}{N} \|, \| \frac{\Delta v_i}{g_i \cdot N} - \frac{\Delta v_i}{N} \|\}\ \forall S_i \in \{S_1, \ldots, S_N\}$ depending on whether $S_i \in K$, or not. Additionally, simple calculations show that $\sigma^2 \geq \sum_{i=1}^{N} E[\|y_i\|^2]$, as required by the Vector Bernstein's Inequality, yields $\sigma^2 \geq \sum_{i=1}^{N} \frac{\|\Delta v_i\|^2}{N^2 \cdot g_i} - \sum_{i=1}^{N} \frac{\|\Delta v_i\|^2}{N^2}$.

Using Inequality 2 we partially satisfy Requirement 2, since we have not yet discussed how B, σ and, thus, ε can be set a priori. In Section 5 we will choose a sampling function providing an ε that is upper bounded by a constant value known to each S_i before a monitoring phase begins. Based on this, we can fully satisfy Requirement 2.

Monitoring Scheme and Efficiency Requirement. Based on Lemma 1, sampled sites need to monitor $Conv(\{e + \frac{\Delta v_i}{g_i}\} : \forall S_i \in K)$ where (i) the estimation $\hat{v}$ of v lies, as Lemma 1(c) shows, and (ii) the true global average v is expected to lie since $E[\hat{v}] = v$. For ease of exposition, we assume that $\hat{v} = v$ and we will loosen this assumption later in this section. In order to track $Conv(\{e + \frac{\Delta v_i}{g_i}\} : \forall S_i \in K)$ according to the existing GM framework, each $S_i \in K$ would need to construct local hyperspheres of the form $B(e + \frac{1}{2}\frac{\Delta v_i}{g_i}, \frac{1}{2}\|\frac{\Delta v_i}{g_i}\|)$, with the union of these local hyperspheres covering the convex hull that encompasses $\hat{v}$. However, these hyperspheres are larger (by a factor of $1/g_i$) than the ones mentioned in our algorithmic sketch. Let us now examine the reason for this important difference.

Compared to the basic GM method (Section 3.1), the above scheme omits hyperspheres of sites that do not get sampled, thus reducing the monitored area. On the other hand, since $g_i \leq 1$, the hyperspheres $B(e + \frac{1}{2}\frac{\Delta v_i}{g_i}, \frac{1}{2}\|\frac{\Delta v_i}{g_i}\|)$ possess larger radii than the $B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$ used in basic GM, and the centers of the spheres are also different. As an example, Figure 3(a) depicts the area that needs to be monitored, which corresponds to the balls of sites S_2 and S_3 covering the shaded part of Figure 3(a), according to Lemma 1. Hence, the current scheme may on one hand reduce FP decisions due to the fact that it uses fewer Δv_i vectors in its convex hull, but on the other hand it may also cause more FP synchronizations because it constructs larger spherical constraints centered at different positions as shown in Figure 3(a). In other words, it may perform better than GM, but this is in no way guaranteed. Obviously, this violates Requirement 1. The following lemma builds on Lemma 1, and demonstrates how our sampling-based monitoring scheme can be validly modified to abide by Requirement 1.

LEMMA 2. *Provided that* $\hat{v} = v$,

$$\hat{v} \in Conv(\{e + \Delta v_i\} : \forall S_i \in K)$$

PROOF. See Appendix □

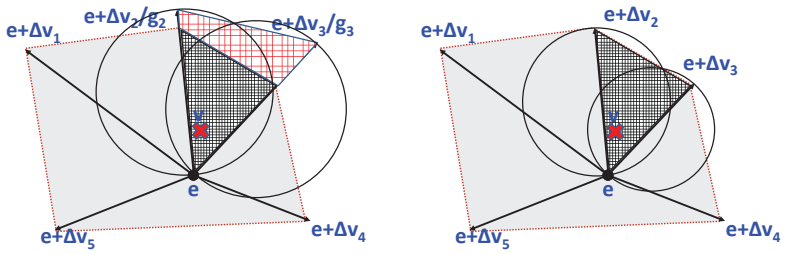
Lemma 2 shows that for the sampled sites to monitor the position of $\hat{v} = v$, they need to construct local constraints of the form $B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$ as we described in our algorithmic sketch. These balls possess the same center and radius as those in the initial GM scheme of Section 3.1 and we can choose g_i such that $|K| \ll N$. Hence, $\bigcup_{S_i \in K} B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|) \subseteq \bigcup_{i=1}^{N} B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$. Figure 3(b) depicts the improvement that Lemma 2 yields in the construction of local balls especially compared to the local constraints induced by Lemma 1 in Figure 3(a). Hence, the new local constraints adhere to Requirement 1 and no additional FPs can be provided by the sampling based scheme. Therefore, Lemma 2 was actually employed in our algorithmic sketch presented at the beginning of this section.

We now focus on removing the $\hat{v} = v$ assumption. Even if $\hat{v} = v$ does not hold, from Inequality 2, we know that with high probability, at least $1 - \delta$, $v \in B(\hat{v}, \varepsilon)$. When $v \in B(\hat{v}, \varepsilon)$, the worst case scenario during the monitoring process appears when $\hat{v}$ is located on the periphery of $\bigcup_{S_i \in K} B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$. Then, the fact that $\hat{v}$ lies on the edge of some GM sphere, combined with the fact that $v \in B(\hat{v}, \varepsilon)$, guarantees that the largest distance v may travel outside the union of the spheres is ε. The first option to handle this situation is to expand the radius of the balls inscribed by sites to $B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\| + \varepsilon)$ so that with probability $1 - \delta$ they include $v \in B(\hat{v}, \varepsilon)$. However, if we do that, Requirement 1 is no longer satisfied. The second option is to allow such an error, which may, however, lead to a FN decision. We opt for the second option and focus on its effect on the FN rate.

Satisfying the Tunable Accuracy Requirement. According to our algorithmic sketch and our analysis so far, a FN decision may occur in the following mutually exclusive cases:
(a) During the distributed monitoring process, upon judging potential threshold crossings of local hyperspheres that were not expanded by an ε factor (as previously described). We consider two sub-cases: In subcase (a1), the local constraint $B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$ of every site $S_i \in K$ has a minimum distance from the threshold surface larger than ε. Subcase (a2) covers the case when the above condition does not hold for at least one $S_i \in K$.
(b) During the synchronization process, where the coordinator probes the sample and uses $B(\hat{v}, \varepsilon)$ to determine if a full synchronization is necessary.

Note however, that these types of FNs cannot occur simultaneously since case (b) can happen only when a threshold crossing is de-

[1] Note that $\sigma^2 \geq \sum_{i=1}^{N} E[\|y_i\|^2] \geq \sum_{i=1}^{N} E[\|y_i\|^2] - (E[\|y_i\|])^2 = \sum_{i=1}^{N} Var[\|y_i\|]$, therefore σ^2 bounds the sum of individual length variances.



(a) Sampling-Based tracking according to Lemma 1, violating Requirement 1 ($S_2, S_3 \in K$)

(b) Requirement 1 abiding sampling-based scheme in Lemma 2 ($S_2, S_3 \in K$)

Figure 3: Sampling-based monitoring over distributed data streams. Shaded areas belong to the convex hull that needs to be monitored according to Lemma 1 and Lemma 2, respectively. The gray area corresponds to the convex hull that is formed by the whole set of sites in the network (N=5).

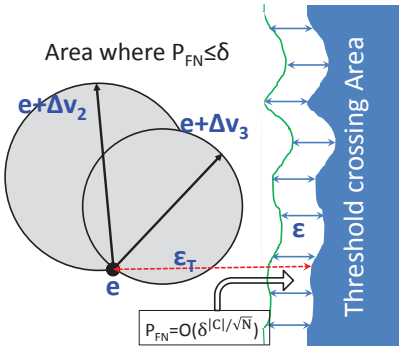


Figure 4: P_{FN} wrt the distance from the threshold surface. In the white (left) area $P_{FN} \leq \delta$ since no ball approaches the surface more than ε.

tected during the monitoring process. We set out our discussion from case (a), which is more complicated.

Case (a1). If $v \in B(\hat{v}, \varepsilon)$, whenever every S_i's $\in K$ local constraint $B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$ has a minimum distance from the threshold surface larger than ε as shown in Figure 4, our choice of not expanding these spheres to $B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\| + \varepsilon)$ does not affect the quality of the monitoring process. This is true because even if v lies outside the monitored area, with high probability $1 - \delta$ it has not changed side with respect to T because $v \in B(\hat{v}, \varepsilon)$. Therefore, when the minimum distance of the union of balls of sampled sites from the threshold surface is larger than ε (see Fig. 4), we cannot have a FN decision unless $v \notin B(\hat{v}, \varepsilon)$. The latter has a probability of δ and $P_{FN} \leq \delta$.

Case (a2). We now examine how the P_{FN} probability is bounded when there exists at least one site $S_i \in K$ with $B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$ placed closer to the threshold surface than ε. Looking at Figure 4, this corresponds to the zone around the threshold surface marked with an ε and corresponding double arrows. If there exists $S_i \in K$ with $B(e + \frac{1}{2}\Delta v_i, \frac{1}{2}\|\Delta v_i\|)$ entering the ε-zone in Figure 4, this means that v is likely to have crossed the threshold surface despite $v \in B(\hat{v}, \varepsilon)$. Let ε_T (red, dotted line in Fig. 4) denote the minimum distance of e from the threshold surface, computed once during a full synchronization process and kept until the upcoming one. Simple calculations show that if no site S_i has a $\|\Delta v_i\| > \varepsilon_T$, the global average cannot have switched side with respect to the threshold surface and no FN decision can occur. Hence, for a FN decision to occur we need *at least one* sampled site to enter the ε-zone and there should exist a number (*at least one*) of sites in the network that have drifted more that ε_T distance from e and *are not* included in K. If at least one of the threshold crossing sites is sampled then a local violation will be detected and no FN can occur at this stage. Therefore, assuming that at a given time point $|C|$ sites cross the threshold, $P_{FN} \leq \prod_{S_i \in C} (1 - g_i)$, since a FP will occur when none of these $|C|$ sites is included in the sample. Fortunately, as we are going to show in the upcoming section, for a properly constructed g_i, even in case that for some sites their drift vectors enter the ε-zone, P_{FN} has an upper bound proportional to $\delta^{\frac{|C|}{\sqrt{N}}}$ which decreases exponentially with the number of threshold crossing sites. Moreover, as we show in Section 5, this bound on P_{FN} is pessimistic, as it is computed on the pathological case where for all $S_i \in C$, $||\Delta v_i|| = \varepsilon_T$. What happens in practice, because v is the average of the drift vectors, is that in order for v to cross the threshold surface, the threshold surface is crossed by either several moderate in length drift vectors (in which case $|C|$ is large), or by fewer but larger drift vectors. In the latter case, we show in Section 5 that the sampling probability of such sites is larger, making it less likely that they will all be omitted from the sample.

Case (b). Please recall that in our algorithmic sketch we mentioned that during a synchronization the coordinator, in its effort to reduce the cost of a potential FP decision, first attempts to save communication by broadcasting a message and requiring only the sampled sites to contribute their Δv_i vectors in order to compute $\hat{v}$. It then checks $B(\hat{v}, \varepsilon)$ for threshold crossing before allowing a full synchronization. In this phase, an erroneous FN decision may occur only when $v \notin B(\hat{v}, \varepsilon)$ which happens with probability at most δ and thus $P_{FN} \leq \delta$.

Based on Inequality 2, we set $\varepsilon = (1 + \sqrt{\ell n(1/\delta)}) \cdot \sigma$. In the next section we provide a sampling function that upper bounds σ by a constant value and tune ε according to the application defined δ. Having bound σ, we showed in this section that P_{FN} can be also bounded by δ. Thus, Requirement 3 is satisfied as well. In the next section we further exhibit that based on the constructed g_i, δ also successfully tunes the sample cardinality $|K|$ and, thus, the anticipated savings of the sampling-based scheme in terms of FP reduction and bandwidth preservation.

5. SETTING THE SAMPLING FUNCTION

In sampling-based geometric monitoring, each S_i individually decides whether to include itself in K or not, using a sampling function $0 \leq g_i \leq 1$. Our sampling-based scheme can accommodate any g_i that samples the $d-$dimensional local vectors of sites. However, not all functions yield desired properties for our scheme. We next present in a step-by-step fashion the elements of a suitable g_i and reason about our choices based on the properties that these elements attribute to our scheme. We eventually derive a proper g_i that simultaneously (a) ensures a sample of $O(\ell n(1/\delta)\sqrt{N})$ size, (b) upper bounds σ and, thus, ε by an a priori (before acquiring the sample) constant value controlled by δ. In that, g_i allows the sampling-based scheme to comply with Requirement 2, (c) given the previous upper bound that determines the size of the $\varepsilon-$zone (Fig. 4), g_i tunes the probability of false negatives (Requirement 3).

• $\|\Delta v_i\|$ **should be included in the nominator of** g_i**.** According to our algorithmic sketch in Section 4.2, upon a local violation the coordinator probes only sites $S_i \in K$ and checks $B(\hat{v}, \varepsilon)$ for threshold

crossing in order to call for a full synchronization, or not. To inscribe $B(\hat{v},\varepsilon)$, the radius ε should obtain (be bounded by) a constant value. In order to come up with a constant value for ε, according to Inequality 2, we need to bound σ. In Section 4.2, we showed that $\sigma^2 \geq \sum_{i=1}^{N} \frac{\|\Delta v_i\|^2}{N^2 \cdot g_i} - \sum_{i=1}^{N} \frac{\|\Delta v_i\|^2}{N^2}$. Apart from g_i, the only variable term included in the latter inequality is the size of the drift $\|\Delta v_i\|$. To eliminate this variable term, $\|\Delta v_i\|$ should be included in the nominator of g_i (calculations follow in the upcoming Inequality 3).

• **$\ell n(1/\delta)$ needs to be included in the nominator of g_i.** As mentioned in Requirement 3, ε should be tunable by (a function of) δ, i.e., $\varepsilon = \varepsilon(\delta)$, so as to allow the size of the $\varepsilon-$ zone to be controlled by the application. Hence, $\ell n(1/\delta)$ needs to be included in the nominator of g_i. Due to the presence of $(1+\sqrt{\ell n(1/\delta)}) \cdot \sigma$ $(= \varepsilon)$ in Inequality 2, we will later show in Equation 4 that placing $\ell n(1/\delta)$ in the nominator of the function allows the application to express the size of the $\varepsilon-$zone as a *fraction* (< 1) of the bound of the approximation error between $\hat{v}$ and v.

• **A term N^x, $x > 0$ is needed in the denominator of g_i.** To ensure communication savings and reduced monitored area, we need $|K| \ll N$. The expected sample cardinality of our scheme is given by $\sum_{i=1}^{N} g_i$. Since this sum iterates over all the N terms, to ensure $|K| \ll N$, we need a term N^x in the denominator of g_i in order to obtain an expected communication cost of $O(N^{1-x})$. What is then required is to compute a proper value for $x > 0$.

• **A constant U such that $U > h \cdot \|\Delta v_i\|$, $h > 1$ is necessary in the denominator of g_i.** Having required that $\|\Delta v_i\|$ lies in the nominator of g_i, to achieve $O(N^{1-x})$ cardinality, the presence of a constant U such that $U > h \cdot \|\Delta v_i\|$ for some $h > 1$ in the denominator of g_i is necessary as well. For example, in a setup where sites receive ± 1 updates per dimension [39, 19, 20] over a sliding window of w size, the maximum $\|\Delta v_i\|$ that may occur is equal to $U = \sqrt{d} \cdot w$. In case of unbounded inputs, a generalization of the bound used in [19] would suffice. In particular, [19] focuses on linear functions and assumes that an estimation of the global count (in one dimension) is available beforehand. It thus sets U equal to that total count estimation. The equivalent in our multidimensional scenario is to utilize e, i.e., the last known global average estimation, and express U as a function of its norm $\|e\|$.

Summarizing, all our previous remarks are satisfied upon setting

$$g_i = \frac{\ell n(1/\delta) \cdot \|\Delta v_i\|}{U \cdot N^x}$$

The expected communication cost is a tunable (using δ) fraction of N, proportional only to $\sum_{i=1}^{N} g_i = O(\ell n(1/\delta) \cdot N^{1-x})$. We then seek for a proper value for $x > 0$. Recalling Vector Bernstein's Inequality (Inequality 2) and using the above g_i, for σ we receive:

$$\begin{aligned}
&\sum_{i=1}^{N} E[\|y_i\|^2] = \sum_{i=1}^{N} \frac{\|\Delta v_i\|^2}{N^2 \cdot \frac{\|\Delta v_i\| \cdot \ell n(1/\delta)}{U \cdot N^x}} - \sum_{i=1}^{N} \frac{\|\Delta v_i\|^2}{N^2} = \\
&\sum_{i=1}^{N} \frac{\|\Delta v_i\|}{N \cdot \sqrt{N}} \frac{U \cdot N^x}{\sqrt{N} \cdot \ell n(1/\delta)} - \sum_{i=1}^{N} \frac{\|\Delta v_i\|^2}{N^2} \cdot \sum_{i=1}^{N} \left(\frac{1}{\sqrt{N}}\right)^2 \overset{Cauchy-Schwarz\ Inequality\ [3,37]}{\leq} \\
&2 \sum_{i=1}^{N} \frac{\|\Delta v_i\|}{N \cdot \sqrt{N}} \frac{U \cdot N^x}{2 \cdot \ell n(1/\delta) \sqrt{N}} - \left(\sum_{i=1}^{N} \frac{\|\Delta v_i\|}{N \cdot \sqrt{N}}\right)^2 = \\
&- \left(\sum_{i=1}^{N} \frac{\|\Delta v_i\|}{N \cdot \sqrt{N}}\right)^2 + 2 \sum_{i=1}^{N} \frac{\|\Delta v_i\|}{N \cdot \sqrt{N}} \frac{U \cdot N^x}{2 \cdot \ell n(1/\delta) \sqrt{N}} \pm \left(\frac{U \cdot N^x}{2 \cdot \ell n(1/\delta) \sqrt{N}}\right)^2 \overset{identity}{=} \\
&- \left(\sum_{i=1}^{N} \frac{\|\Delta v_i\|}{N \cdot \sqrt{N}} - \frac{U \cdot N^x}{2 \cdot \ell n(1/\delta) \sqrt{N}}\right)^2 + \left(\frac{U \cdot N^x}{2 \cdot \ell n(1/\delta) \sqrt{N}}\right)^2 \Leftrightarrow
\end{aligned}$$

$$\sum_{i=1}^{N} E[\|y_i\|^2] \leq \left(\frac{U \cdot N^x}{2 \cdot \ell n(1/\delta) \sqrt{N}}\right)^2 = \sigma^2 \tag{3}$$

In our analysis we made use of the Cauchy-Schwarz Inequality [3, 37], which states that if $a_1, \ldots, a_N$ and $b_1, \ldots, b_N$ are two sequences of real numbers, then $\left(\sum_{i=1}^{N} a_i \cdot b_i\right)^2 \leq \sum_{i=1}^{N} a_i^2 \cdot \sum_{i=1}^{N} b_i^2$. The equality holds if and only if the sequences are proportional. In order to express ε as a fraction of U and as a function of δ, while at the same time avoiding an undesirable dependence on the network scale N, we pick $x = 1/2$. Then, $\varepsilon = (1+\sqrt{\ell n(1/\delta)}) \cdot \sigma = (\frac{1+\sqrt{\ell n(1/\delta)}}{2 \cdot \ell n(1/\delta)}) \cdot U$, while B can be set to $B = \frac{\|\Delta v_i\|}{N \cdot g_i} = \frac{U}{\ell n(1/\delta) \cdot \sqrt{N}}$. Notice that the choice of $\ell n(1/\delta)$ in the nominator of g_i is the lowest value that we could use in order to obtain a tunable by δ increase in the expected communication cost, while at the same time being able to express ε as a percentage of U. The latter claim is true due to the fact that $(\frac{1+\sqrt{\ell n(1/\delta)}}{2 \cdot \ell n(1/\delta)}) < 1, \forall \delta < 1/e$ (i.e., a range that contains the typical values for δ).

The Sampling Function. Pointing out the characteristics of our sampling-based *GM* scheme, for given $\delta \in (0, 1/e)$:

$$\begin{cases}
g_i = \frac{\|\Delta v_i\| \ell n(1/\delta)}{U \cdot \sqrt{N}} & \text{Sampling Function} \\
\varepsilon = (\frac{1+\sqrt{\ell n(1/\delta)}}{2 \cdot \ell n(1/\delta)}) \cdot U & \hat{v}\text{ Estimation Error} \\
\sum_{i=1}^{N} g_i = O(\ell n(1/\delta) \sqrt{N}) & \text{Expected Sample Size}
\end{cases} \tag{4}$$

Notice that $g_i < 1$ and $\varepsilon \leq \sigma^2/B$ practically holds, as required by the Vector Bernstein's Inequality. In addition, ε is controllable using the δ parameter. Furthermore, note that when δ decreases, then ε also decreases, while the expected sample size increases logarithmically. This is a trade-off between bandwidth consumption and accuracy that our sampling-based scheme achieves by a single, application defined parameter δ.

Notice that the proposed g_i does not explicitly impose a lower bound on sample size. For example, if all sites have very small Δv_i vectors, then their sampling probabilities will be small. However, even if none of them gets sampled (such a case becomes less likely as the number of sites increases), our algorithm will estimate - according to Estimator 1 - that $\hat{v} = e$. That is, the current position of v coincides with the estimate vector e (the last known, due to synchronization, global average vector), leading to the conclusion that the function has not crossed the threshold. In any such case, according to our analysis using the Vector Bernstein Inequality, our estimation is accurate within ε from the true average with (controllably) high probability. On the other hand, our framework needs to ensure that it samples enough sites when the global vector v does cross the threshold surface, so as to avoid FNs. In our work (see Lemma 3 below), we bound P_{FN} based on both the number of threshold crossing sites and the distance of the spheres from the $\varepsilon-$ zone (and, thus, from the threshold surface).

EXAMPLE 2. *Recalling Example 1, we note that common relative humidity values in a server room range between 0.4 and 0.6 (i.e, 40% and 60%) rH. In such cases, the maximum $\|\Delta v_i\|$ may occur when the reading of a sensor gradually shifts from 0.4 to 0.6 (or vice versa), yielding a maximum $\|\Delta v_i\| = \sqrt{(0.6^2 - 0.4^2)^2 + (0.6-0.4)^2} \approx 0.28$. Thus, $U = 0.28$. The following table computes, for $N = 100$ and $N = 961$, the values of ε, the range of g_i values in this example, and $\ell n(1/\delta)\sqrt{N}$ (an upper*

bound on $\sum_{i=1}^{N} g_i$*) for* δ *values of 0.1 and 0.05. We note that this upper bound becomes smaller compared to N as N increases. Moreover, note that smaller* δ *values result in smaller* ε *and larger* g_i *values, as smaller* δ *values point to a requirement for fewer FNs.*

δ	N	$\sqrt{N}$	*Range of* g_i	ε	$\ell n(1/\delta)\sqrt{N}$
0.1	*100*	*10*	*[0,0.23]*	*0.15*	*24*
0.1	*961*	*31*	*[0,0.08]*	*0.15*	*72*
0.05	*100*	*10*	*[0,0.3]*	*0.13*	*30*
0.05	*961*	*31*	*[0,0.097]*	*0.13*	*93*

Completing the puzzle for P_{FN} bounds. In Section 4.2, we mentioned that when some of the sampled sites enter the ε-zone, $P_{FN} \leq \prod_{S_i \in C}(1-g_i)$, where C the set of threshold crossing sites. Under the light of the chosen g_i we receive $P_{FN} \leq \prod_{S_i \in C}(1 - \frac{\|\Delta v_i\| \ell n(1/\delta)}{U \cdot \sqrt{N}})$. However, we also noted that for a site $S_i \in C$, $\|\Delta v_i\| > \varepsilon_T$, since otherwise S_i cannot have crossed the threshold surface. Therefore, since $g_i < 1$, substituting above:

$$P_{FN} \leq (1 - \frac{\ell n(1/\delta) \cdot \varepsilon_T}{U \cdot \sqrt{N}})^{|C|} \leq e^{-\frac{|C| \cdot \ell n(1/\delta) \cdot \varepsilon_T}{U \cdot \sqrt{N}}} = \delta^{\frac{|C| \cdot \varepsilon_T}{U \cdot \sqrt{N}}}$$

As mentioned in Section 4.2, this bound on P_{FN} is a worst-case bound that is derived from a pathological case, in which for all $S_i \in C$, $||\Delta v_i|| = \varepsilon_T$. However, what happens in practice, because v is the average of the drift vectors, is that in order for v to cross the threshold surface, the threshold surface is crossed (i) by either several moderate in length drift vectors, in which case $|C|$ is large and P_{FN} is small, since it decreases exponentially with the number of threshold crossing sites, or (ii) by fewer but larger drift vectors that, thus, have larger sampling probabilities. In the latter case, it is less likely that they are all omitted from the sample.

Thus, apart from ensuring $P_{FN} \leq \delta$ when no local constraint enters the ε-zone as discussed in Section 4.2, we also bounded the complementary case, and note that P_{FN} may become even lower than δ when $|C|$ is sufficiently larger than $\sqrt{N}$. We emphasize that the minimum distance of e from the threshold surface, i.e., ε_T (see Fig 4), is computed during a synchronization and is, thus, a known parameter until the next central data collection. In any case, the size of the $\varepsilon-$zone can be tuned to the desirable extend using δ as discussed above. Based on the above discussion and our analysis in Section 4.2 the following lemma comes naturally.

LEMMA 3. *The Sampling-Based GM Scheme being set according to Equation 4 yields:*

- $P_{FN} \leq \delta$ *if* $\forall S_i \in K, B(e + \frac{\Delta v_i}{2}, \frac{\|\Delta v_i\|}{2}) \bigcap \varepsilon - zone = \emptyset$
- $P_{FN} = O(\delta^{\frac{|C|}{\sqrt{N}}})$ *otherwise*

where C denotes the set of threshold crossing sites. Thus, one can properly tune δ *to obtain the desired FN probability.*

6. EXPERIMENTS

We develop a simulation environment in Java in order to evaluate the performance of our techniques. We compare the communication cost (number of messages) as well as the number of FP and FN synchronization decisions of our sampling-based scheme, henceforth referred to as **SGM**, against other GM related techniques that have been proposed in the literature. More precisely:

- The Geometric Monitoring framework of Section 3.1 introduced in [35], termed **GM**.
- The GM framework enhanced with the balancing optimization presented in [35], termed **BGM**. In BGM, the coordinator tries to probe a subset of Δv_is that may partially cancel out the crossing ones due to their different direction. If such a subset exists, it knows that $v(t)$ has not crossed the threshold without requiring a full synchronization. Please refer to [35] for more details.
- The Prediction-Based Geometric Monitoring Framework, and in particular the CAA technique proposed in [16, 17], henceforth referred to as **PGM**. We adopt a Velocity-Acceleration predictor and present the best performance PGM shows upon varying the window according to which predictions are formed from 3 to 10 measurements (roughly the amount of stories received hourly).

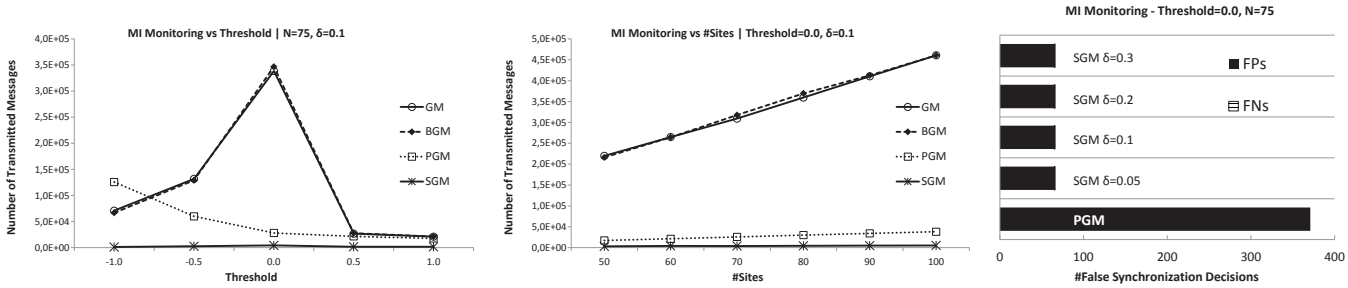
We emphasize that the BGM and the PGM approaches are orthogonal to our sampling-based GM framework. Despite this fact, to better perceive the benefits of our novel SGM approach and expose its features, in our experimentation we form a worst case scenario for SGM by not applying any orthogonal approach on top of it. Moreover, note that BGM and PGM are not orthogonal to each other, since the CAA approach [16, 17] switches among monitoring models instead of balancing their drift vectors.

Data Sets. We utilize two real world datasets. The first dataset is the Reuters Corpus (RCV1-v2) [26] data, termed *Reuters*, also used in related work [35, 36, 16]. It is composed of 804414 records of news stories which have been categorized and have been tagged with a list of terms (features). As previous works [35, 36, 16], we focus on tracking the terms *Febru, Ipo, Bosnia* and their co-occurrences with the Corporate/Industrial category. We monitor the relevance between a (term, category) pair using two different functions: the Mutual Information (MI) function and the χ^2 function (please refer to [16, 17] for more details). We use a sliding window of 200 documents for this dataset which is roughly the amount of news stories received daily [16, 17]. Since, the number of records in the data is limited, we use the Reuters data to evaluate our algorithms in mediocre distributed settings of size $N = 50$ to $N = 100$ and provide some initial comparisons with related work [35, 16, 17] where no more than 100 sites are considered.

The second dataset, termed *Jester* [18], contains 4.1 million ratings between -10 and 10 on 100 jokes from 73421 users. We use this dataset to approximately monitor the sum in buckets of equi-width histograms of the above rating range, based on L_∞ distance as well as the Jeffrey Divergence (JD) [31]. More precisely, we use these functions to measure the distance (cost) of encoding the current global histogram at each time instance, to the one communicated during the last central data collection. In addition to L_∞ and JD, the third function we experiment on (in the Appendix), is the tracking of the Self-Join (SJ) size [17, 15, 8] (essentially the L_2 Norm) of the vector hosting the expected counts in the aforementioned histogram buckets. Since users provide ratings for 100 distinct cases, we utilize a sliding window of 100 observations for this dataset. Regarding the degree of distribution, we vary N between 100 and 1000.

Metrics and Parameter Settings. Throughout our study, for each (dataset, function) pair we initially measure the number of communicated messages varying the value of the threshold, i.e., T, keeping the number of sites to a fixed value that lies in the middle ($N = 75$ for Reuters and $N = 500$ for Jester) of the aforementioned distribution ranges. This helps demonstrate the performance of the candidate schemes upon altering the placement of the threshold surface.

Then, for the middle threshold case we investigate the communication cost for increasing network scales. In the Jester dataset where larger network sizes can be tested, we also investigate the cost of messages per site (instead of just the total number of mes-



(a) MI Monitoring: Comm. Cost vs Threshold (b) MI Monitoring: Comm. Cost vs Sites (c) MI Monitoring: Sensitivity to δ

Figure 5: Reuters Data Set: Performance of our Techniques on *MI* Monitoring

sages), as this gives an indication on how the cost of each site scales when the number of sites increases. For large numbers of sites and for constrained applications such as sensor networks, where an increased number of transmitted messages results in reduced network lifetime, this per site cost should ideally remain steady (or increase at a small rate) as we scale to larger networks.

For SGM, we also vary the posed tolerance constraint i.e., the application defined probability δ, between 0.05 and 0.3 with a default value of 0.1 and perform a sensitivity analysis on the number of false (FP, FN) synchronization decisions in comparison with the FPs produced by the best (at each case) of the competitors.

Main Findings. Our experimental analysis demonstrates that our sampling-based SGM method significantly reduces the number of transmitted messages and the number of false positives, with the benefits becoming more profound when the number of sites increases. Please note that, since in GM each false positive results in communication from all sites and the coordinator, significantly reducing the number of false positives translates into a corresponding bandwidth reduction (or energy consumption reduction, in case of applications with sensor sites) on *each* site. This is validated by the scalability experiment performed on the (larger) Jester dataset, where the corresponding benefits *per site*, compared to GM, increase in larger network setups. Finally, the false negatives of SGM are in all cases lower than the specified tolerance parameter δ.

6.1 Reuters Dataset Monitoring

We first focus on the Reuters dataset. Figure 5(a) and Figure 5(b) present the communication cost of GM, BGM, PGM and SGM varying the value of the posed threshold and the scale of the distributed network for the default value of $\delta = 0.1$ when monitoring the Mutual Information function. We present results for the Febru term, as the trends are similar for the other two (Bosnia, Ipo) terms.

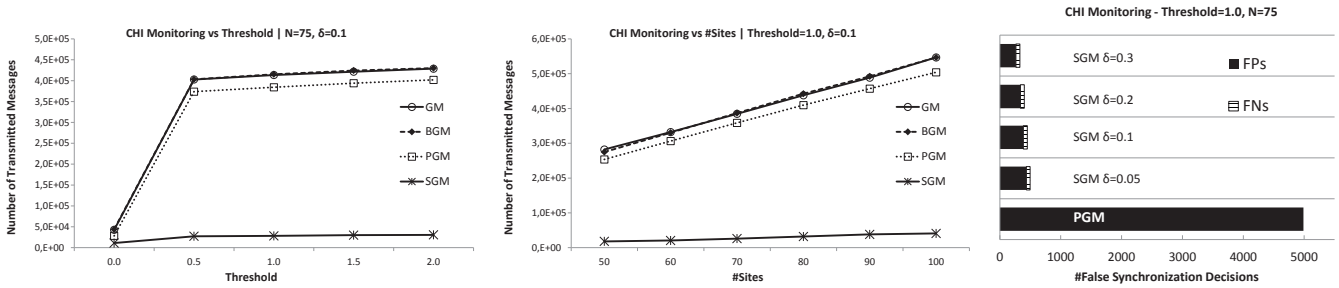
In both figures, the lines corresponding to GM, BGM almost coincide, showing that in this data set the balancing optimization does not reduce the communication cost. The reason is that, when many sites cross the threshold surface moving towards similar directions, an additive effect on their Δv_is naturally exists. Therefore, the coordinator probes almost all of the non-crossing sites so as to balance the added drift. On the other hand, the prediction-based mechanism of PGM reduces the communication cost by more than an order of magnitude compared to GM and BGM for different threshold values when $N = 75$. Still, our proposed sampling-based scheme SGM (star-marked line approaching the x axis in the figure) performs from 6 times (for $T = 0$) to two orders of magnitude (for $T = -1$) better than PGM across various thresholds, with at least an order of magnitude improvement for the rest of the tested threshold values. Despite its good performance, PGM performs worse than GM and BGM for $T = -1$. This is because, contrary to our sampling-based scheme, the prediction-based mechanism PGM can practically achieve (as shown in [16, 17]), but does not guarantee communication savings over the baseline solution of GM.

Moreover, in Figure 5(b) our SGM approach performs from 5 to almost an order of magnitude times better than PGM, which in turn outperforms GM and BGM for the fixed $T = 0$, across the various network scales. The benefits of SGM increase with the number of sites as, not only does it reduce the number of FPs (also depicted in Figure 5(c)), but also in each FP it requires transmission from $O(\sqrt{N})$ sites, in comparison to $O(N)$ sites for the other techniques.

Figure 5(c) presents a sensitivity analysis on the effect of δ to the number of FP, FN decisions. Recall that FPs are responsible for the unnecessary portion of communicated messages. As a result, our sensitivity analysis also exposes the trade-off among bandwidth consumption caused by FPs and accuracy in terms of FNs for SGM. The horizontal bars depict the number of FP decisions of PGM (which, as we just showed, outperforms GM and BGM) compared to the FP and FN decisions of SGM, under different δ values ranging between 0.05 and 0.3. FPs and FNs for each given δ are drawn in stacked bars as explained by the corresponding legends, while the overall length of the bars represents the total number of false decisions. As shown by the figure, the communication reduction that was observed in Figure 5(a) and Figure 5(b) for $T = 0$ is translated to a corresponding ratio (of about 6) of the FP decisions of PGM over those of SGM with $\delta = 0.1$. Moreover, we can observe that the number of FP decisions of SGM remains stable across different δ values, while the SGM framework in practice produces no FNs for this (function, dataset) pair.

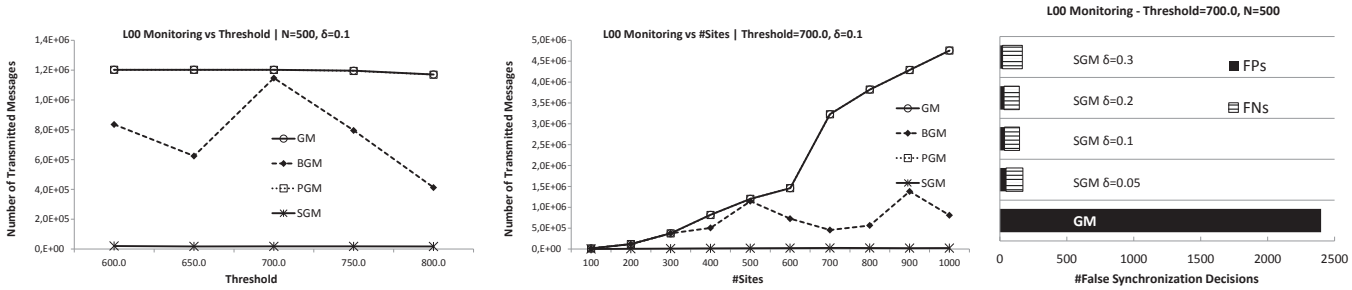
Turning our interest to the χ^2 case, the main difference in Figure 6(a) and Figure 6(b) is that PGM performs only slightly (<0.2 times) better than BGM and GM. Monitoring χ^2 results in a monitoring process at a higher dimensional space (which in turns means trying to accurately predict more components of local and predicted global vectors). This affects the quality of the predictions and the size of the monitored convex hulls in PGM, therefore its limited improvements. Apart from the above remark, SGM reduces the bandwidth consumption from 3.8 times for $T = 0$, to more than an order of magnitude compared to the other candidates for the rest of threshold values (Fig. 6(a)). In addition, SGM needs more than one order of magnitude (between 13 and 16 times) fewer messages than its competitors across different network scales (Fig. 6(b)).

We now focus on Figure 6(c) and the amount of FPs, FNs depicted there. SGM ensures more than an order of magnitude reduction on the amount of false decisions (represented by the total length of the stacked bar of FP, FN counts) compared to the second best alternative of PGM. Figure 6(c) demonstrates that increasing



(a) χ^2 Monitoring: Comm. Cost vs Threshold (b) χ^2 Monitoring: Comm. Cost vs Sites (c) χ^2 Monitoring: Sensitivity to δ

Figure 6: Reuters Data Set: Performance of our Techniques on χ^2 Monitoring



(a) L_∞ Monitoring: Comm. Cost vs Threshold (b) L_∞ Monitoring: Comm. Cost vs Sites (c) L_∞ Monitoring: Sensitivity to δ

Figure 7: Jester Data Set: Performance of our Techniques on L_∞ Monitoring

δ causes FP decisions to be reduced by more than 15% in each bar of the histogram. FNs slightly increase with increased δ values, but are very rare in all cases. The reduction in FP with increased δ values is easily understood, since the expected sample size is proportional to $\ell n(1/\delta)\sqrt{N}$. Thus, increasing δ decreases the sample size and, thus, the monitored space responsible for FPs. Regarding FNs, they are rare in all cases. For instance, for SGM and $\delta = 0.05$, ∼8000 updates arrive per site (see the size of the Reuters data set and $N \leq 100$) and the number of FNs is just 61, which corresponds to a ratio lower than 0.01. SGM typically provides fewer FNs (in this experiment, always by at least a factor of 5) than the posed δ according to which the tolerance to FNs is tuned.

6.2 Jester Dataset Monitoring

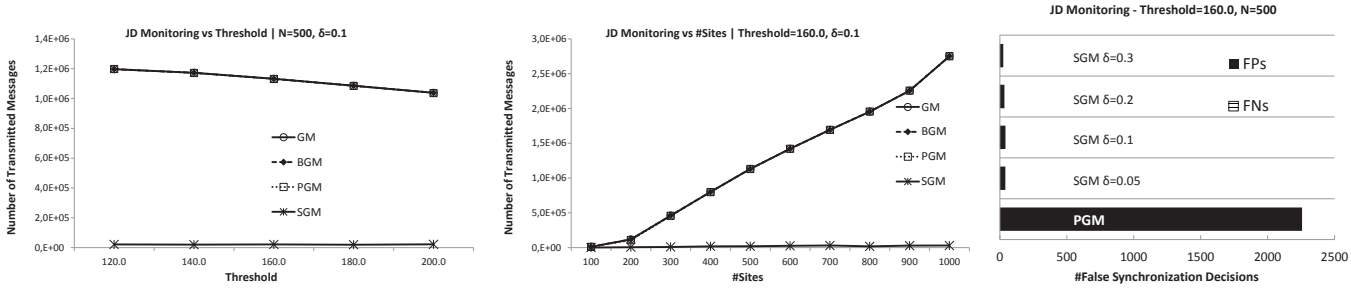
We now focus on the Jester dataset which, due to its larger size, allows us to also perform tests in larger topologies. The default number of sites used in this dataset is 500. Two general observations coming out of Figure 7 and Figure 8 we mention that:

- In this larger scale dataset, the performance of the PGM approach is equivalent to the baseline GM. This validates our claim in Section 2 where we noted that PGM may perform well in small to medium sized network distributions, but increasing the network scale makes the existence of inaccurate predictors in some sites more probable and, thus, PGM becomes more prone to FPs.
- Figure 7 shows that balancing may help reduce the bandwidth consumption in that particular (function, dataset) pair. Nonetheless, in Figure 8 where only the function and the threshold value (surface) is altered, BGM provides no improvements. This comes as no surprise, as BGM adopts a simple heuristic hoping to probe sites with drift vectors of opposite direction compared to the threshold crossing ones. Hence, contrary to SGM, BGM does not guarantee communication reduction. Furthermore, contrary to our proposed SGM approach, none of the BGM or PGM mechanisms provide a way to tune the expected bandwidth consumption according to posed accuracy standards.

Focusing on specific figures, in Figure 7(a) we point out that the bandwidth consumption achieved by our SGM (star-marked line approaching the x axis in the figure) approach is from 25 to 64 times lower than the best alternative (BGM). Moreover, upon varying the network scale between 100 and 1000 sites in Figure 7(b), the communication cost reduction by SGM reaches a factor of 64 for $N = 900$, while constantly being at least 20 times lower compared to BGM, for different degrees of network distribution. Comparing the cost of SGM against PGM or GM, for instance when $N = 1000$ SGM results in 206 times fewer messages, with at least 20 times fewer messages for any other network scale.

Concluding our discussion on L_∞, the sensitivity analysis in Figure 7(c) shows the trade-off among unnecessary bandwidth consumption due to FP and accuracy in terms of FN decisions for different δs. SGM is compared to the FPs in GM, as BGM causes full synchronization only progressively, thus FPs cannot be counted in a distinct manner. As this figure shows, the number of FP decisions tends to be reduced upon increasing δ, while the number of FNs tends to increase, both of which are the expected behavior. Given that about 4850 updates arrive in every site of the network using the Jester dataset, the 148 FNs for $\delta = 0.3$ correspond to a ratio of just 3% false negatives, while the corresponding ratio for the ∼110-120 FNs and for the rest of the examined δ values never exceeds 2.3%. Hence, once again the amount of FNs is less than the posed δ.

In Figure 8 we focus on the Jeffrey Divergence monitoring function. In Figure 8(a) and Figure 8(b), all three competitive techniques (GM, BGM and PGM) exhibit comparable performances.



(a) JD Monitoring: Comm. Cost vs Threshold (b) JD Monitoring: Comm. Cost vs Sites (c) JD Monitoring: Sensitivity to δ

Figure 8: Jester Data Set: Performance of our Techniques on Jeffrey Divergence Monitoring

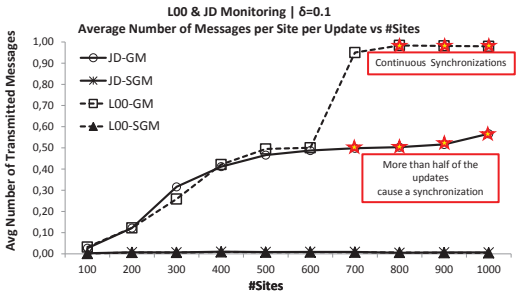


Figure 9: Average Number of Messages Transmitted by *each Site* per Data Update

Our SGM framework reduces the consumed bandwidth up to a factor of 56 across different thresholds for $N = 500$ and the communication gains progressively approach two orders of magnitude by increasing the network scale to $N = 1000$ sites (Fig. 8(b)). Regarding the number of false synchronization decisions and the sensitivity on the chosen δ, Figure 8(c) exposes the absence of FNs and the reduction of FPs by about 20% as we increase δ above 0.1.

Due to space constraints, the case of Self-Join size monitoring is summarized in Section 6.4 and then described in the Appendix.

6.3 Messages Per Site in Large Networks

In order to validate our claim regarding scalability to larger network distributions and to resource constrained environments, apart from measuring the total communication cost (number of messages) transmitted in the network, we further study the average number of messages transmitted by *each site*. That is, we measure the average number of messages a site transmitted per each update of its data. An average value close to 1 indicates that each site in the network transmitted a message after each data update, which is equivalent to a synchronization process.

Figure 9 presents the average number of transmissions per site and data update for the GM and SGM schemes in L_∞ and in Jeffrey Divergence monitoring when varying the size of the network scale. Figure 9 shows that increasing the scale in GM (and the other alternatives that have similar performance to GM in Fig. 7(b) and Fig. 8(b)) results in a continuous increase in the number of transmitted messages per site. This is more evident in L_∞ monitoring where, starting at 800 sites, GM behaves as the naive choice of continuous central data collection, since at least one site exhibits a local violation, which results in communication by all other sites as well. In Jeffrey Divergence monitoring this effect is less pronounced until $N = 500$, but still each site transmits a message in over half of its data updates for larger network sizes. On the contrary, the SGM approach is very slightly affected by the increase in network distribution, since the number of sampled sites increases with the logarithm of the network size. Thus, the benefits of SGM not only increase with larger network topologies, but it is also more appropriate for resource constrained environments, such as battery-powered sensor networks, where it is desirable to reduce the amount of communication per site in order to prolong the network lifetime.

6.4 Additional Results

In the Appendix we provide the following additional results: (a) we provide statistics on the duration of FNs, showing that even if FNs occur (in a controllable manner), the missed threshold crossings are detected soon afterwards in the future, most often in the next synchronization decision; (b) we include the analysis of Self Join size monitoring for which SGM provides more than an order of magnitude fewer transmissions compared to GM, PGM. BGM can fall short compared to our SGM up to 8 times, but typically provides between 2-3 times worse communication cost in terms of the number of transmitted messages; (c) we compare our SGM approach using the proposed g_i (Section 5) with a variant that uses the SGM framework but naively samples sites with equal probability, i.e., performs Bernoulli sampling. What we show is that SGM outperforms the Bernoulli sampling variant across different network scales with gains reaching a factor of 50.

7. CONCLUSIONS

In this work we rendered the GM framework of [35] capable to operate in highly distributed settings. We initially exhibited the culprits that cause the GM approach to become impractical due to severe scalability issues. To encounter these issues, we introduced a novel sampling-based geometric monitoring technique capable of performing the tracking process utilizing only a sample of the available sites. The sample size entailed by our methods is proportional to $\sqrt{N}$ and also dependent on application's accuracy requirements. Our experimental evaluation shows that our sampling-based techniques can significantly reduce the communication cost throughout the monitoring process with controllable accuracy guarantees, outperforming other competitors proposed in the literature.

8. REFERENCES

[1] S. Burdakis and A. Deligiannakis. Detecting outliers in sensor networks using the geometric approach. In *Proc. of ICDE Conference*, pages 1108–1119, 2012.

[2] E. J. Candès and Y. Plan. A probabilistic and ripless theory of compressed sensing. *CoRR*, abs/1011.3854, 2010.

[3] A. L. Cauchy. *Cours d'analyse de l'École royale polytechnique*. Paris, France, 1821.

[4] G. Cormode. The continuous distributed monitoring model. *SIGMOD Rec.*, 42(1):5–14, may 2013.

[5] G. Cormode and N. Duffield. Sampling for big data: A tutorial. In *Proc. of ACM SIGKDD Conference*, pages 1975–1975, 2014.

[6] G. Cormode and M. Garofalakis. Sketching streams through the net: Distributed approximate query tracking. In *Proc. of VLDB Conference*, pages 312–323, 2005.

[7] G. Cormode and M. Garofalakis. Streaming in a connected world: querying and tracking distributed data streams. In *Proc. of SIGMOD Conference*, pages 1178–1181, 2007.

[8] G. Cormode and M. Garofalakis. Approximate continuous querying over distributed streams. *ACM Transactions on Database Systems*, 33(2):9:1–9:39, 2008.

[9] G. Cormode, M. Garofalakis, S. Muthukrishnan, and R. Rastogi. Holistic aggregates in a networked world: distributed tracking of approximate quantiles. In *SIGMOD*, pages 25–36, 2005.

[10] G. Cormode, S. Muthukrishnan, and K. Yi. Algorithms for distributed functional monitoring. *ACM Trans. Algorithms*, 7(2):21:1–21:20, mar 2011.

[11] C. Cranor, T. Johnson, O. Spataschek, and V. Shkapenyuk. Gigascope: A stream database for network applications. In *ACM SIGMOD*, pages 647–651, 2003.

[12] M. Dilman and D. Raz. Efficient reactive monitoring. In *INFOCOM*, volume 2, pages 1012–1019, 2001.

[13] M. Gabel, A. Schuster, and D. Keren. Communication-efficient distributed variance monitoring and outlier detection for multivariate time series. In *IEEE IPDPS*, pages 37–47, 2014.

[14] M. Garofalakis, J. Gehrke, and R. Rastogi. Querying and mining data streams: You only get one look a tutorial. In *Proc. of SIGMOD Conference*, pages 635–635, 2002.

[15] M. Garofalakis, D. Keren, and V. Samoladas. Sketch-based geometric monitoring of distributed stream queries. *In VLDB*, 6(10):937–948, 2013.

[16] N. Giatrakos, A. Deligiannakis, M. Garofalakis, I. Sharfman, and A. Schuster. Prediction-based geometric monitoring over distributed data streams. In *SIGMOD*, pages 265–276, 2012.

[17] N. Giatrakos, A. Deligiannakis, M. Garofalakis, I. Sharfman, and A. Schuster. Distributed geometric query monitoring using prediction models. *ACM Trans. Database Syst.*, 39(2):16:1–16:42, may 2014.

[18] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Inf. Retr.*, 4(2):133–151, 2001.

[19] Z. Huang, K. Yi, Y. Liu, and G. Chen. Optimal sampling algorithms for frequency estimation in distributed data. In *INFOCOM*, pages 1997–2005, 2011.

[20] Z. Huang, K. Yi, and Q. Zhang. Randomized algorithms for tracking distributed count, frequencies, and ranks. In *Proc. of PODS*, pages 295–306, 2012.

[21] S. R. Jeffery, M. Garofalakis, and M. J. Franklin. Adaptive cleaning for rfid data streams. In *Proceedings of the 32Nd International Conference on Very Large Data Bases*, VLDB '06, pages 163–174. VLDB Endowment, 2006.

[22] M. Kamp, M. Boley, D. Keren, A. Schuster, and I. Sharfman. Communicationefficient distributed online prediction by dynamic model synchronization. In *ECML PKDD*, pages 623–639, 2014.

[23] R. Keralapura, G. Cormode, and J. Ramamirtham. Communication-efficient distributed monitoring of thresholded counts. In *SIGMOD*, pages 289–300, 2006.

[24] D. Keren, G. Sagy, A. Abboud, D. Ben-David, A. Schuster, I. Sharfman, and A. Deligiannakis. Geometric monitoring of heterogeneous streams. *Knowledge and Data Engineering, IEEE Transactions on*, 26(8):1890–1903, 2014.

[25] A. Lazerson, I. Sharfman, D. Keren, A. Schuster, M. Garofalakis, and V. Samoladas. Monitoring distributed streams using convex decompositions. *Proc. VLDB Endow.*, 8(5):545–556, jan 2015.

[26] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(Apr):361–397, 2004.

[27] Z. Liu, B. Radunović, and M. Vojnović. Continuous distributed counting for non-monotonic streams. In *Proc. of PODS*, pages 307–318, 2012.

[28] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tinydb: an acquisitional query processing system for sensor networks. *ACM Trans. Database Syst.*, 30:122–173, March 2005.

[29] A. Manjhi, V. Shkapenyuk, K. Dhamdhere, and C. Olston. Finding (recently) frequent items in distributed data streams. In *ICDE*, pages 767–778, 2005.

[30] O. Papapetrou and M. N. Garofalakis. Continuous fragmented skylines over distributed streams. In *IEEE ICDE*, pages 124–135, 2014.

[31] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision*, 40(2):99–121, 2000.

[32] G. Sagy, D. Keren, I. Sharfman, and A. Schuster. Distributed threshold querying of general functions by a difference of monotonic representation. *In VLDB*, 4:46–57, 2010.

[33] G. Sagy, I. Sharfman, D. Keren, and A. Schuster. Top-k vectorial aggregation queries in a distributed environment. *J. Parallel Distrib. Comput.*, 71(2):302–315, 2011.

[34] C.-E. Särndal, B. Swensson, and J. Wretman. *"Model Assisted Survey Sampling"*. Springer-Verlag New York, Inc. (Springer Series in Statistics), 1992.

[35] I. Sharfman, A. Schuster, and D. Keren. A geometric approach to monitoring threshold functions over distributed data streams. In *SIGMOD*, pages 301–312, 2006.

[36] I. Sharfman, A. Schuster, and D. Keren. Shape sensitive geometric monitoring. In *PODS*, pages 301–310, 2008.

[37] J. M. Steele. *The Cauchy-Schwarz Master Class: An Introduction to the Art of Mathematical Inequalities, Chap.1,*. Cambridge University Press, New York, NY, USA, 2004.

[38] M. Tang, F. Li, and Y. Tao. Distributed online tracking. In *Proc. of SIGMOD Conference*, pages 2047–2061, New York, NY, USA, 2015.

[39] Q. G. Zhao, M. Ogihara, H. Wang, and J. J. Xu. Finding global icebergs over distributed data sets. In *PODS*, 2006.

APPENDIX

A. PROOFS OF LEMMA 1 AND LEMMA 2

LEMMA 1. *For Estimator 1 the following hold:*
(a) Estimator 1 is an unbiased estimator of v when sampling $\forall S_i \in \{S_1, \ldots, S_N\}$ *with* $0 \leq g_i \leq 1$.
(b) $E[\hat{v}] \in Conv(\ e+\Delta v_1, \ldots, e+\Delta v_N)$
(c) $\hat{v} \in Conv(\{e+\frac{\Delta v_i}{g_i}\} : \forall S_i \in K)$

PROOF.
(a) To prove that the estimator is unbiased we need to show that $E[\hat{v}] = v$. Recall from Equation 1 that $\sum_{S_i \in K} \frac{\Delta v_i}{g_i} = \sum_{i=1}^{N} \Delta' v_i$ and that $E[\Delta' v_i] = g_i \cdot \frac{\Delta v_i}{g_i} + (1-g_i)\cdot 0 = \Delta v_i$. By applying the properties of the expected value we get:

$E[\hat{v}] = E[e + \frac{\sum_{i=1}^{N} \Delta' v_i}{N}] = e + \frac{\sum_{i=1}^{N} E[\Delta' v_i]}{N} = e + \frac{\sum_{i=1}^{N} \Delta v_i}{N} = v$
(b) Obvious, since $E[\hat{v}] = v$ and $v \in Conv(\ e+\Delta v_1, \ldots, e+\Delta v_N)$.
(c) $\hat{v}$ is a convex combination of the $e+\Delta' v_i$ vectors and, therefore, lies in their convex hull. Since $\Delta' v_i = 0$ for all sites not included in the sample, the convex hulls $Conv(\{e+\frac{\Delta v_i}{g_i}\} : \forall S_i \in K)$ and $Conv(\{e+\Delta' v_i\} : \forall i \in [1,N])$ coincide. □

LEMMA 2. *Provided that* $\hat{v} = v$,

$$\hat{v} \in Conv(\{e+\Delta v_i\} : \forall S_i \in K)$$

PROOF. Lemma 1(c) we shows $\hat{v} \in Conv(\{e+\frac{\Delta v_i}{g_i}\} : \forall S_i \in K)$, and we already know that $v \in Conv(\{e+\Delta v_i\} : \forall S_i \in \{S_1, \ldots, S_N\})$. But, since $\hat{v} = v$, also $\hat{v} \in Conv(\{e+\Delta v_i\} : \forall S_i \in \{S_1, \ldots, S_N\})$ should hold. Combining the above:

$$\hat{v} \in Conv(\{e+\Delta v_i\} : \forall S_i \in \{S_1, \ldots, S_N\}) \cap Conv(\{e+\frac{\Delta v_i}{g_i}\} : \forall S_i \in K).$$

Let X_N denote the above intersection of the convex hulls when the N sites perform sampling trials. To prove the lemma it suffices to show that X_N is equivalent to $Conv(\{e+\Delta v_i\} : \forall S_i \in K)$.

Before we begin, we emphasize that in case a site gets sampled, $g_i > 0$ must hold, otherwise it cannot be included in the sample. Let $[y;w]$ denote the set of all convex combinations among $y, w \in \mathbb{R}^d$, i.e., $[y,w] = \{\forall q \in \mathbb{R}^d : q = \lambda \cdot y + (1-\lambda)\cdot w\ , 0 \leq \lambda \leq 1\}$. The proof will be derived inductively, in each of the steps abusively specifying K_N to denote a sample obtained out of N sites. Note beforehand that any intersection of convex sets is convex.

At the base case, $N = 1$ site exists, $Conv(e+\Delta v_1) = [e; e+\Delta v_1]$. If $S_1 \in K_1$, $Conv(e+\frac{\Delta v_1}{g_1}) = [e; e+\frac{\Delta v_1}{g_1}]$. Multidimensional points $e+\Delta v_1$, $e+\frac{\Delta v_1}{g_1}$ possess the same starting point e and differ only in the scale among Δv_1, $\frac{\Delta v_1}{g_1}$. Therefore, e, $e+\Delta v_1$, $e+\frac{\Delta v_1}{g_1}$ are collinear and since $g_1 \leq 1$, $[e; e+\Delta v_1] \cap [e; e+\frac{\Delta v_1}{g_1}] = [e; e+\Delta v_1]$ $\Rightarrow X_1 = [e; e+\Delta v_1] = Conv(\{e+\Delta v_i\} : \forall S_i \in K_1)$. On the other hand, if $S_1 \notin K_1$, $X_1 = e$. So, the lemma holds for $N = 1$.

We assume that the lemma holds for $N-1$ sites, i.e.,

$$X_{N-1} = Conv(\{e+\Delta v_i\} : \forall S_i \in \{S_1, \ldots, S_{N-1}\}) \cap Conv(\{e+\frac{\Delta v_i}{g_i}\} : \forall S_i \in K_{N-1}) = Conv(\{e+\Delta v_i\} : \forall S_i \in K_{N-1})$$

and examine the addition of S_N.

By adding $e+\Delta v_N$ to $Conv(\{e+\Delta v_i\} : \forall S_i \in \{S_1, \ldots, S_{N-1}\})$ we not only added the convex combinations $[e; e+\Delta v_N]$ as in the case where $N = 1$, but also all other possible convex combinations from any $y \in Conv(\{e+\Delta v_i\} : \forall S_i \in \{S_1, \ldots, S_{N-1}\})$ to the newly introduced points $[e; e+\Delta v_N]$ i.e., $[y; [e; e+\Delta v_N]]$. The same holds for $Conv(\{e+\frac{\Delta v_i}{g_i}\} : \forall S_i \in K_{N-1})$. Having said that, the candidate points for addition in X_{N-1} to form X_N are included in the intersection of the newly introduced convex combinations. That is, the set of points that X_N will add (compared to X_{N-1}) belong to the intersection of all possible convex combinations from every point included in X_{N-1} to the new $[e; e+\Delta v_N]$, $[e; e+\Delta' v_i]$: $\forall z \in X_{N-1}$ $[z; [e; e+\Delta v_N]] \cap [z; [e; e+\Delta' v_i]]$. If $S_N \notin K_N$, $[z; [e; e+\Delta v_N]] \cap [z; e] = [z; e] \Rightarrow X_{N-1} = X_N = Conv(\{e+\Delta v_i\} : \forall S_i \in K_N)$. On the other hand, when $S_N \in K_N$, taking into consideration the collinearity among e, $e+\frac{\Delta v_N}{g_N}$ and $e+\Delta v_N$, $[z; [e; e+\Delta v_N]] \cap [z; [e; e+\frac{\Delta v_N}{g_N}]]$ $= [z; [e; e+\Delta v_N]]$. Eventually, X_N=$X_{N-1} \cup [z; [e; e+\Delta v_N]] =$ $Conv(\{e+\Delta v_i\} : \forall S_i \in K_N)$, which completes the proof. □

B. JESTER DATASET - SJ MONITORING

In the Jester dataset we approximately monitor the sum in buckets of equi-width histograms, based on L_∞ distance as well as the Jeffrey Divergence (JD) [31]. In addition to L_∞ and JD, here we also experiment with a third function tracking the Self-Join (SJ) size [17, 15, 8] (essentially the L_2 Norm), of the vector hosting the expected counts in histogram buckets. Figure 10 presents the bandwidth consumption achieved by SGM compared to the rest of the candidates across different thresholds, network scales and values of δ. According to Figure 10(a) and Figure 10(b), SGM may require more than an order of magnitude fewer message transmissions compared to (GM, PGM) across different thresholds and network scales. SGM performs from 2 (for $T = 210$) to 8 (for $T = 180$) times better than BGM upon varying the threshold. Additionally, for network scales up to 400 sites, SGM reduces the transmitted messages up to a factor of 7 compared to BGM, while for higher amounts of distribution SGM mostly halves the number of messages BGM requires.

On the other hand, the sensitivity analysis of Figure 10(c) shows and the amount of FPs is reduced by more than an order of magnitude compared to GM for the tested tolerance δ values (recall that BGM progressively probes sites, thus FPs cannot be counted in a distinct manner). The same figure shows that increasing δ causes FP decisions to be reduced by more than 10% when switching from $\delta = 0.05$ to $\delta = 0.1$ and by almost 20% from $\delta = 0.1$ to $\delta = 0.2$. At the same time, FNs marginally increase with increased δ. Once again, the reduction in FPs with increased δ values is explained by the fact that the expected sample size is proportional to $\ell n(1/\delta)\sqrt{N}$. Thus, increasing δ decreases the sample size and the tracked area. Regarding FNs, their percentage lies below the corresponding δ value in each experiment of Figure 10(c).

Eventually, Figure 11 presents the average number of messages transmitted per site and data update for the GM and SGM schemes. It is not difficult to see that even for mediocre network scales of 100 sites, more than half of the updates caused a synchronization in GM, while this percentage exceeds 80% for topologies with 800 or more sites.

C. DURATION OF FNS

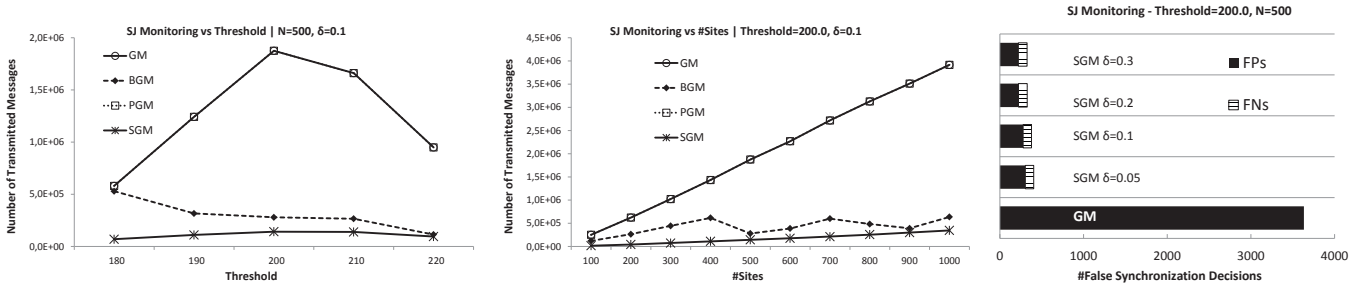
A discriminating fact among FP and FN synchronization decisions lies on the nature of their effect. FP decisions have an instant effect as the coordinator becomes aware of a FP, and the certain overhead on the consumed bandwidth it caused, by the end of a synchronization. Contrary to FPs, a FN decision has both the instant effect of saving bandwidth while it should not, as well as a persistent effect. In particular, upon a FN occurrence and for a number

	Threshold					
	0,5		1		1,5	
#Sites	Mode	Mdn	Mode	Mdn	Mode	Mdn
60	1	3	1	3	1	2
70	1	4	1	3	1	2
80	1	3	1	3	1	3
90	2	3	1	4	1	2
100	2	3	1	2	1	1

Table 2: FN Duration - χ^2 Monitoring

	Threshold					
	190		200		210	
#Sites	Mode	Mdn	Mode	Mdn	Mode	Mdn
600	2	2	1	1	1	1
700	1	1	1	1	1	1
800	1	1	1	1	2	1
900	1	1	1	1	1	1
1000	1	1	1	3	1	1

Table 3: FN Duration - *SJ* Monitoring



(a) *SJ* Monitoring: Comm. Cost vs Threshold (b) *SJ* Monitoring: Comm. Cost vs Sites (c) *SJ* Monitoring: Sensitivity to δ

Figure 10: Jester Data Set: Performance of our Techniques on $Self-Join$ size Monitoring

of time units, the application continuous to consider that the monitored function lies on one side of T, while $f(v(t))$ has switched side. This misconception is held by the coordinator until either a synchronization (FP or not), or $v(t)$ again switches to its initial side with respect to T. Consequently, we enhance our study by concentrating on the anticipated duration of a FN decision, indicatively providing results for χ^2 and *SJ* monitoring. In our study we report holistic aggregates and in particular the Mode and Median (denoted by Mdn) statistics for FN duration.

As both Table 2 and Table 3 demonstrate, the most frequent situation is the one where our proposed SGM approach compensates the coordinator for a FN decision immediately after its occurrence, i.e., the corresponding duration is 1 time unit. This is expressed by the Mode=1 value in the vast majority of the cases cited in the corresponding tables. On the other hand, interpreting the cited median values, we can observe that most of the times SGM needs no more than 3 time units to compensate for a FN apparition for χ^2 (Table 2), while needing 1 time unit (i.e., Mode=Mdn) for SJ (Table 3).

Overall, we can safely conclude that even when SGM does produce FN decisions (recall that JD and MI were practically FN free) it possesses the ability to immediately compensate the tracking process for them. This is due to the fact that for low δ values, the probability of missing the event of a threshold crossing in consecutive time units decreases with the number of time units.

D. COMPARISON WITH A BERNOULLI SAMPLING VARIANT

As we described in Section 2, the limitations (see points (a)-(d) in the last paragraph of Section 2) incorporated in [39, 19, 20, 27] do not allow them to become valid g_i choices for SGM. The question that naturally arises is what if we reside to a simpler g_i, instead of the one proposed in Section 5, which uses the SGM framework but naively samples sites with equal probability, i.e., performs Bernoulli sampling. For a fair comparison with our techniques, in case of this Bernoulli sampling variant each site's g_i is set to $\ell n(1/\delta)/\sqrt{N}$ yielding analogous expected sample size $(O(\ell n(1/\delta)\sqrt{N}))$ as the function that we proposed in Section 5. Please note that the Bernoulli sampling variant still utilizes optimizations that we proposed in this paper, such as the observation that sampled sites do not need to scale their Δv_i vectors by $1/g_i$.

We compare SGM incorporating the g_i of Section 5 (as in all previous evaluations), with the Bernoulli sampling variant in terms of the number of transmitted messages for different network scales. Figure 12 presents the respective comparison pairs for each monitored function (L_∞, *JD*, *SJ*) in the Jester dataset. Pairwise comparisons shown in the figure include SGM's performance marked with the respective function abbreviation (e.g. L_∞-SGM), against the respective variant (e.g., L_∞-Bernoulli).

According to Figure 12 we observe the following: (a) in SJ monitoring, SJ-Bernoulli performs 2-3 times worse than our proposed SJ-SGM across the examined network scales, (b) in Jeffrey Di-

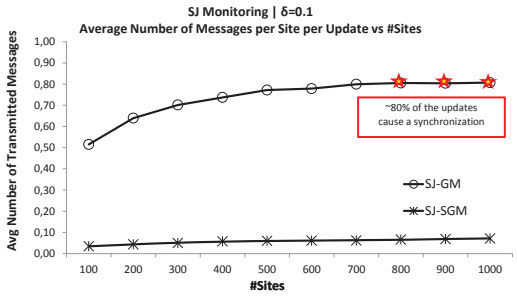


Figure 11: Average Number of Messages Transmitted by *each Site* per Data Update for SJ Monitoring

vergence monitoring, JD-Bernoulli falls short from 6 to 36 times compared to our JD-SGM and (c) $L_\infty-$Bernoulli provides from 5 to 50 times more transmitted messages than our $L_\infty-$SGM proposal. These ratios exhibit the ability of our proposed g_i to decrease communication burden compared to other, straightforward, sampling function choices. The differences in the performance with the Bernoulli sampling variant are mainly attributed to the fact that, contrary to the g_i we proposed in this work (Section 5), Bernoulli sampling does not take into consideration the size of the local deviation vector $\|\Delta v_i\|$. Thus, sites with small deviations that less affect the global average but lie near the threshold surface, are equally probable to be included in the sample as peers with large $\|\Delta v_i\|$ that push the global average away from it. A plausible characteristic is that such a behavior is not allowed by our proposed sampling function which incorporates $\|\Delta v_i\|$ in its calculation formula.

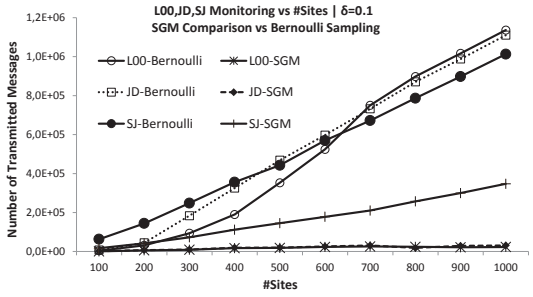


Figure 12: Comparison of SGM vs Bernoulli Sampling Variant