IBM Research Report

Accuracy Analysis of Data Aggregation for Network Monitoring

Nikoletta Sofra

Department of Electrical and Electronic Engineering
Imperial College
London, UK

Ting He, Petros Zerfos, Bong Jun Ko, Kang-won Lee
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598, USA

Kin K. Leung
Electrical and Electronic Engineering and Computing Departments
Imperial College
London, UK

Accuracy Analysis of Data Aggregation for Network Monitoring

Nikoletta Sofra, Ting He, Petros Zerfos, Bong Jun Ko, Kang-won Lee and Kin K. Leung

Abstract— **The quality of computing certain aggregation functions based on incomplete measurements for the purpose of distributed network monitoring is considered. Network monitoring plays a fundamental role in network management systems by providing timely information on the network status, which is crucial for network administration purposes. To reduce network overhead and for easier assimilation, this information is usually presented by calculating a few key aggregate metrics. The aggregates are periodically computed from a large number of detailed events collected continuously during the course of the network operations. Under errors induced by network delays, the accuracy of typical aggregation functions used in network management systems (e.g., sum, average, maximum) is evaluated both analytically and by simulations. The results provide a quantifiable trade-off between accuracy and timeliness of the information acquired, which can then be used to design and optimize network management systems.**

Index Terms—**Network Monitoring, Data Aggregation, Quality of Management**

I. INTRODUCTION

NETWORK monitoring is a critical function used for the efficient and robust operation of modern networks, through which timely information on the status of network elements and communication links is provided to network administrators. Network monitoring systems continuously collect numerous measurements of key performance indicators and events such as bandwidth utilization and link status and process them into more meaningful, aggregate metrics such as

Nikoletta Sofra is with the Department of Electrical and Electronic Engineering, Imperial College, London UK (phone: +44 (0)207 59 46310; fax: +44 (0)207 594 6302; e-mail: nikoletta.sofra06@imperial.ac.uk).

Ting He, Petros Zerfos, Bong Jun Ko and Kang-won Lee are with the IBM T.J. Watson Research Center, Hawthorne, NY, USA (e-mail: {the, pzerfos, bongjun_ko, kangwon}@us.ibm.com)

Kin Leung is with Electrical and Electronic Engineering and Computing Departments, Imperial College, London UK (e-mail: kin.leung@imperial.ac.uk)

average delay and network availability. The results are of aggregation are then presented to network operators for further diagnosing potential service problems and choosing corrective actions. Aggregate metrics also form the quantitative basis around which service level agreements (SLAs) between network providers and their customers are usually structured [4][5], as compliance with (or deviation from) agreed-upon levels can be readily assessed. For this reason, accurate computation of aggregate metrics becomes a particularly important issue for any network monitoring system and the topic of study for this work.
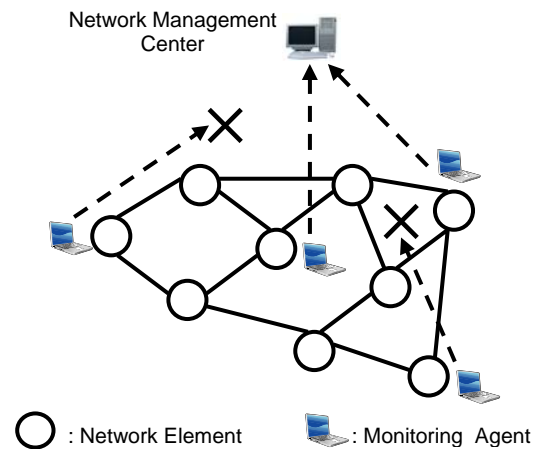

Figure 1: Processing center and monitoring agents of a distributed network management system

The accuracy of computed aggregates is affected by errors introduced by the measurement process as well as the transmission of the collected data to the processing station. The latter case occurs when monitoring of the network is distributed to several monitoring agents and processing of the measured data takes place in a remote network management center (as illustrated in Figure 1). Such deployment can cause transmission delays and packet loss of sampled data, which skew the results of the periodic calculation of aggregate metrics. Furthermore, depending on the size of the network and the network monitoring architecture, the computation of the aggregate metric itself might take place distributedly, with partial results transferred for use over several levels of aggregation, a process that introduces further errors in the calculations.

Recent work on computing in-network aggregates in the field of sensor networks [9][12][13][14][19] primarily focuses on the trade-off between accuracy and energy consumption for extending the lifetime of the network under certain topological assumptions regarding the distribution of computation (e.g. tree-based, hierarchical, centralized, etc). Results from studies on monitoring of network variables [15][16][17] explore the trade-off between communication overhead and accuracy of aggregation.

This work provides an analysis of the error that is induced in the calculation of the typical aggregation functions used for network monitoring, when the measurement samples are delayed or missing, *independent* of the particular topology used for delivering the (partial) results, or the communication costs for transmitting the underlying measurements. As such, it provides a characterization of the *quality of aggregation* irrespective of the underlying network monitoring architecture, which extends the concept of Quality of Management (QoM) introduced in [1]. More specifically, this work makes the following contributions:

- An error analysis of aggregation functions used for network monitoring based on a stochastic model for describing the generation and transmission of measured events.
- Simulation results that provide insights on how the error is affected by network parameters such as communication delay and the rate of event generation, as well as design parameters such as period of aggregation.

The rest of this paper is organized as follows: Section 2 discusses the system model and the basic assumptions, which the accuracy analysis of aggregation functions in Section 3 is based on. Section 4 presents simulation results that verify the analysis. Related work is overviewed in Section 5 and the paper concludes in Section 6.

II. EVENTS, MODELS AND QUALITY OF AGGREGATION

In order to analyze the error of aggregation functions in the presence of delayed or missing measurements, we need to define a model for the generation of network events that are monitored by the network monitoring system. In this section, we first outline a generic model for the network monitoring function and event generation and then define the quality of aggregation.

A. Event Generation in Network Monitoring Systems

Network administrators are faced with a multitude of unpredictable situations during the course of the operation of network such as network outages, sudden traffic spikes, device and application misconfiguration, as well as issues caused by third-party peering networks. Information about these situations may be manifested as *events*, which typically include a number of characteristics such as the time the situation arose, a value that describes the intensity of the situation, indication of managed resources affected by the situation, etc. These events, along with measurements on various performance indicators such as bandwidth utilization, link delay, dropped sessions, etc.

are collected by the network monitoring system, for further processing, archiving and presentation to the network administrators.

In large network management deployments, collection of the events and measurements is performed distributedly by monitoring agents that are placed close to the managed network elements and report the measured samples to a centralized network management center (Figure 1). There, calculation of user-defined metrics takes place, which typically involves the use of aggregation functions such as SUM, AVERAGE, COUNT, and MIN/MAX. Then, the results are presented to a network administrator for decision making. In the remainder of this paper, we examine the behavior of typical aggregation functions used in the computation of user-defined performance metrics at the network management center, as captured events might be delayed or lost during their transmission from the monitoring agents to the network management center (Figure 1). More details on events and the model used hereafter can be found in [1].

B. Discrete Event Model

For the analysis of the effect of the network delay on the accuracy of the results obtained through processing using aggregation functions, we assume the following event model based on stochastic processes:

- There are N managed network elements in the network, represented by a set $\{1, 2, …, N\}$, and each such network element continuously collects information by means of a monitoring agent.
- The j-th event captured by a monitor i is represented by $(x_i(j), t_i(j))$, where $x_i(j)$ is the intensity of the event and $t_i(j)$ is an increasing sequence of the time representing the time the event occurs where $t_i(j) > t_i(j-1)$.
- The events captured at the monitor i occur according to a homogeneous Poisson process with arrival rate λ_i. A special case of the event arrival process is that the arrival rates *at each* monitor is homogeneous, in which case we denote by λ the arrival rate at each monitor, i.e., $\lambda_i = \lambda$.
- The intensity of an event $x_i(j)$ at a monitor i is drawn from a random variable X_i. We assume $\{X_i\}$, i=1, …,N, are i.i.d. random variables, for which $E[X_i] = m_X$ and $\text{Var}(X_i) = \sigma_X^2$.
- Whenever a monitoring agent captures such an event, it sends it to the network management center, where it can be used for aggregation (Figure 2). The network delay is assumed to be a random variable, D_i, exponentially distributed with mean $1/\mu_i$. (A special case is when $\mu_i = \mu$ for all i=1, …, N)
- The network management center aggregates the events that it has received periodically, with a period T. The aggregation takes place at time $t_{agg}=kT+h$ for all the events that have occured within the interval $[(k-1)T, kT)$.

Even though the additional time h facilitates for the delay induced in the network, so that events occurred within the according time interval reach the aggregation point on time, there can still exist some events that miss this deadline. This results in some data missing at the time when the aggregation occurs, introducing an error on these calculations, as illustrated in Figure 2. We assume that events are time stamped, so that those received after the according time interval are dropped and do not interfere with later calculations.
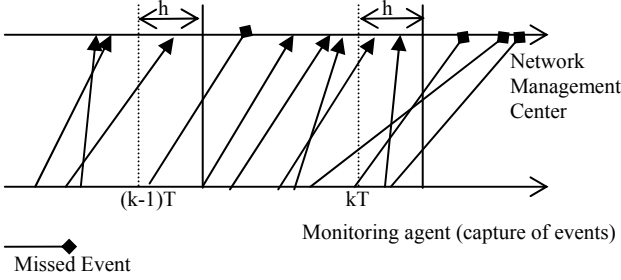

Figure 2 Event Model. Some events arrive at the network management center out of order or too late.

We denote by $S_{i,k}$ the set of events captured by the monitor i in $[(k-1)T, kT)$, by $M_{i,k} \subset S_{i,k}$ the set of events that miss the deadline $kT+h$, and by $L_{i,k} = S_{i,k} - M_{i,k} \subset S_{i,k}$ the set of events that do not miss the deadline and are included in the aggregation. Since the events' arrival process and the network delay in the time interval $[(k-1)T, kT)$ are independent of k, we can omit the index k, by simply denoting the above sets of events by S_i, M_i, and L_i, respectively, for the sake of our analysis. We further denote by S, M, and L the corresponding sets of all events from all monitors, i.e., $S = \bigcup_{i=1}^{N} S_i$, $M = \bigcup_{i=1}^{N} M_i$, and $L = \bigcup_{i=1}^{N} L_i$.

C. Aggregation Functions

Typical aggregation functions used for network monitoring can be classified into summary functions such as count (and histogram), sum, average, variance (and standard deviation), and exemplary functions such as max/min, median, and any percentile. We refer to [20] for the distinction between summary and exemplary aggregation functions. These can be combined with additional preprocessing, e.g., multiplying weights or applying de-duplication (i.e., filtering out duplicates) to the input values, and post-processing including comparing the aggregated values against given thresholds. We hereby categorize these functions in Table I, including the partial results needed as an input for the next level of aggregation, and the type of function that should be used intermediately. It is worth noting that if the aggregation functions are applied only to distinct values (e.g., count distinct values), then an aggregator node should maintain a set of all the distinct values observed, and only the union operator can be used to combine partial aggregates (independent of the final aggregation function).

D. Quality of Aggregation Functions

We focus our study on *accuracy* as the property that primarily characterizes the *quality of aggregation*. The accuracy of a particular aggregation function is captured by looking at the difference between the output of the aggregation function applied to the (ground-truth) set of events S_i and that applied to the included set L_i. Formally, if we define an aggregation function $F(S)$ as an operator that takes a set of events S as the input and outputs a real number, then the quality of F under our event model is measured by *the absolute error*:

$$ERR_F = | F(S) - F(L) |.$$

In the following section, we provide analytical results of the expected value $E[ERR_F]$ and variance $Var(ERR_F)$ for four aggregation functions: CNT (Count), SUM (Sum), AVG (Average), and MAX (Maximum). We will also consider the weighted generalization of some of the above aggregation functions, where different weights are given to the events from different monitors during the aggregation. The weighted function will be denoted by appending a prefix 'W-' to the corresponding function, e.g., W-CNT and W-SUM, etc.

Note that the mean square error, $E[(F(S) - F(L))^2]$ can be also derived from our analysis; since

$$E[(F(S) - F(L))^2] = Var(ERR_F) + E^2[ERR_F],$$

it suffices to analyze the mean and variance of the absolute error.

TABLE I
AGGREGATION FUNCTIONS

	Function	**Partial Results**	**Intermediate Aggr. Function**
Summary	Sum	Sum	summation
	Average	sum, count	summation
	variance, std. dev.	sum of values and squared values, count	summation
	count, histogram	count, histogram	summation
Exemplary	max, min	max, min	max, min
	p-percentile (0<p<1)	data set (incl. identical values)	append (no aggregation)

III. ANALYSIS OF QUALITY OF AGGREGATION

Based on our event model in Section II.B, the expected number and the variance of events occurring at monitor i within a time interval T are:

$$E[| S_i |] = \lambda_i T \tag{1a}$$

and

$$Var(| S_i |) = \lambda_i T . \tag{1b}$$

The probability that an event, monitored and sent at time t will not be aggregated within the appropriate time interval is equal to the probability that the monitored event will not arrive to the aggregation site before the aggregation time $t_{agg}=kT+h$

due to the delay incurred by the network. These events are hereinafter referred to as missed events. The number of missed events within a time interval T is random variable following the Poisson distribution, with mean

$$E[|M_i|] = \lambda_i e^{-\mu_i h} \tfrac{(1-e^{-\mu_i T})}{\mu_i} \qquad (2a)$$

and variance

$$Var(|M_i|) = \lambda_i e^{-\mu_i h} \tfrac{(1-e^{-\mu_i T})}{\mu_i} \qquad (2b)$$

as proven in Appendix I.

Similarly, it can be easily shown that the number of events that are included in the aggregation (received events) follows a Poisson distribution of mean and variance

$$E[|L_i|] = Var(|L_i|) = \lambda_i (T - e^{-\mu h} \tfrac{(1-e^{-\mu T})}{\mu}) \qquad (3)$$

The total expected number of all events, all missed events, and all included events in the network can be calculated by summation over all monitors using equations (1b), (2a) and (3) respectively.

A. Error analysis: COUNT

When the aggregation function is the enumeration of events, the expected absolute error and its variance are equal to those of the number of missed events. Therefore,

$$E[ERR_{CNT}] = E[|M|] = \sum_{i=1}^{N} \lambda_i e^{-\mu_i h} \tfrac{(1-e^{-\mu_i T})}{\mu_i} \qquad (4)$$

and variance

$$Var(ERR_{CNT}) = Var(|M|) = \sum_{i=1}^{N} \lambda_i e^{-\mu_i h} \tfrac{(1-e^{-\mu_i T})}{\mu_i} \qquad (5)$$

If the enumeration is weighted, the way the weights are assigned (and thus their statistical properties) depends on the specifics of the application. In the following we assume that the probability of the event being missed is uncorrelated with the weight value that would be assigned to this event.

For the sake of analysis, we assume that the weight given to event j from the monitor i is a random variable W_{ij} with mean w_i and variance $\sigma_{W_i}^2$, independent of the event but not necessarily identically distributed across monitors. Then the expected absolute error is given by

$$ERR_{W-CNT} = \left| \sum_{i=1}^{N} \sum_{j \in S_i} W_{ij} - \sum_{i=1}^{N} \sum_{j \in L_i} W_{ij} \right|, \qquad (6)$$

and if we assume the weights to be non-negative, we get

$$ERR_{W-CNT} = \sum_{i=1}^{N} \sum_{j \in M_i} W_{ij}, \qquad (7)$$

According to [2], and assuming that events across monitors are uncorrelated we can calculate these moments as follows:

$$E[ERR_{W-CNT}] = \sum_{i=1}^{N} E[|M_i|]E[W_{ij}]$$
$$= \sum_{i=1}^{N} \lambda_i e^{-\mu_i h} \tfrac{(1-e^{-\mu_i T})}{\mu_i} w_i, \qquad (8)$$

$$Var(ERR_{W-CNT}) = Var(\sum_{i=1}^{N} \sum_{j \in M_i} W_{ij})$$
$$= \sum_{i=1}^{N} (E[|M_i|]\sigma_{W_i}^2 + (E[W_{ij}])^2 Var(|M_i|)) \qquad (9)$$
$$= \sum_{i=1}^{N} \lambda_i e^{-\mu_i h} \tfrac{(1-e^{-\mu_i T})}{\mu_i} (\sigma_{W_i}^2 + w_i^2)$$

The results in (8) and (9) are also applicable to the case of the assignment of a constant weight, w_i to each monitor (indicating for example that information received by different monitors is of different importance). In that case, we have $\sigma_{W_i}^2 = 0$.

B. Error analysis: SUM

The SUM aggregation function is defined as the sum of the values (i.e., intensity) of the events, i.e., $SUM(S) = \sum_{j \in S} X_j$.

Then the absolute error of the SUM aggregation function is given by

$$ERR_{SUM} = \left| \sum_{i \in S} X_j - \sum_{j \in L} X_j \right| = \left| \sum_{j \in M} X_j \right|. \qquad (10)$$

Let us assume for now that X_j's are nonnegative, as it is usually the case for the quantities measured in network monitoring (e.g. delay, packet loss rate etc). In this case, we have $ERR_{SUM} = \sum_{j \in M} X_j$, and we can calculate the expected value and the variance of the absolute sum error, treating it as a sample sum of random sample size, as the number of missed events is a discrete random variable [2]. Since the number of missed events $|M|$ is independent of the intensities of those events X_j,

$$E[ERR_{SUM}] = E\left[\sum_{j \in M} X_j \right] = E[|M|]E[X_j]$$
$$= m_X \sum_{i=1}^{N} \lambda_i e^{-\mu_i h} \tfrac{(1-e^{-\mu_i T})}{\mu_i} \qquad (11)$$

and

$$Var(ERR_{SUM}) = E[|M|]\sigma_X^2 + (E[X_j])^2 Var(|M|)$$
$$= (\sigma_X^2 + m_X^2) \sum_{i=1}^{N} \lambda_i e^{-\mu_i h} \tfrac{(1-e^{-\mu_i T})}{\mu_i} \qquad (12)$$

Similarly, with weighted count, the absolute error for the calculation of the weighted sum where weights are i.i.d. for events of the same monitor, and independent (but not identically distributed) for different monitors can be calculated by:

$$ERR_{W-SUM} = \sum_{i=1}^{N} \sum_{j \in M_i} W_{ij} X_i(j), \qquad (13)$$

We can get the expected value and the variance of the absolute error for weighted sum aggregation:

$$E[ERR_{W-SUM}] = \sum_{i=1}^{N} E[|M_i|]E[W_{ij}X_i(j)]$$
$$= \sum_{i=1}^{N} (m_X w_i + Cov(X_i,W_i))\lambda_i e^{-\mu_i h} \frac{(1-e^{-\mu_i T})}{\mu_i},$$
(14)

and

$$Var[ERR_{W-SUM}]$$
$$= \sum_{i=1}^{N} E[|M_i|]Var(X_iW_i) + (E[X_iW_i])^2 Var(|M_i|)$$
$$= \sum_{i=1}^{N} \lambda_i e^{-\mu_i h} \frac{(1-e^{-\mu_i T})}{\mu_i}\left(Var(X_iW_i) + (m_X w_i + Cov(X_i,W_i))^2\right)$$
(15)

where a dependence of the values of the weights on the underlying values, if such exists, can be accommodated. The variance of the product X_iW_i can be estimated as in [3]. Note that the above results can also be used for the case that event values are independent but not identically distributed across monitors (by viewing $W_{ij}X_i(j)$ as the new event value).

In the case where the sums from different monitors are given different fixed weights on the total sum, the weight would not depend on the intensity values of events. Assuming that the weights given to different monitors as well as the values received are independent to each other and to delays,

$$E[ERR_{W-SUM}] = E\left[\sum_{i=1}^{N} w_i \sum_{j\in M_i} X_i(j)\right]$$
$$= m_X \sum_{i=1}^{N} w_i \lambda_i e^{-\mu_i h} \frac{(1-e^{-\mu_i T})}{\mu_i},$$
(16)

and

$$Var(ERR_{W-SUM}) = Var\left(\sum_{i=1}^{N} w_i \sum_{j\in M_i} X_i(j)\right)$$
$$= \sum_{i=1}^{N} w_i^2 Var\left(\sum_{j\in M_i} X_i(j)\right)$$
$$= (m_X^2 + \sigma_X^2)\sum_{i=1}^{N} w_i^2 \lambda_i e^{-\mu_i h} \frac{(1-e^{-\mu_i T})}{\mu_i}$$
(17)

C. Error analysis: AVERAGE

The AVERAGE aggregation function is defined as the sample average of a set of events' values (i.e., intensity), i.e.

$$AVERAGE(S) = \frac{1}{|S|}\sum_{j\in S} X_j.$$

Then the absolute error of the average value is given by

$$ERR_{AVG} = \left|\frac{1}{|S|}\sum_{i\in S} X_j - \frac{1}{|L|}\sum_{j\in L} X_j\right|.$$
(18)

Since $S = M \cup L$, and $M \cap L = \phi$, we can rewrite the above formula by dividing the set of events into missed events (M) and received events (L):

$$ERR_{AVG} = \left|\frac{1}{|S|}\sum_{i\in M} X_j - \left(\frac{1}{|L|} - \frac{1}{|S|}\right)\sum_{j\in L} X_j\right|.$$
$$= \left|\frac{|M|}{|S|}\frac{1}{|M|}\sum_{i\in M} X_j - \frac{1}{|L|}\sum_{j\in L} X_j\right|$$
(19)

The merit of (19) is that given M and L, the two summations are about disjoint sets of events, which have independent values. The computation of the expected value of (19) is complicated by the absolute operator, and thus we try to provide bounds instead. By triangle inequality, we can obtain a trivial lower bound of zero and an upper bound:

$$E[ERR_{AVG}] \le E\left[\frac{|M|}{|S|}\frac{1}{|M|}\sum_{i\in M} X_j\right] + E\left[\frac{|M|}{|S|}\frac{1}{|L|}\sum_{i\in L} X_j\right]$$
$$= \Pr\{miss\}m_X + \Pr\{miss\}m_X$$
$$= 2m_X e^{-\mu h}\frac{(1-e^{-\mu T})}{\mu T}$$
(20)

assuming homogeneous transmission delay. Similar techniques can be used to obtain bounds on the error variance. Specifically, a lower bound of the error variance is given by (details are omitted due to space constraints)

$$Var(ERR_{AVG}) \ge \max\left(0, \sigma_X^2\left(\frac{p}{1-p}\right)\left(\frac{1-e^{-\lambda T}}{\lambda T}\right) - 4m_X^2 p^2\right),$$
(21)

where $p = e^{-\mu h}\frac{(1-e^{-\mu T})}{\mu T}$, the probability that an event misses the deadline, and an upper bound is given by

$$Var(ERR_{AVG}) \le \sigma_X^2\left(1 - \frac{1-e^{-\lambda T}}{\lambda T}\right).$$
(22)

D. Error analysis: MAX and MIN

We present results only for MAX function since MIN function can be analyzed in an identical manner. We assume that the probability density function of the intensity values X_j is known and given by f_X (therefore their cumulative distribution function, F_X is known as well). We can use order statistics to find the distribution of the maximum value among a set of realizations of X, given the size of the set, K:

$$f_{\max(X)|K} = K(F_X(x))^{K-1} f_X(x).$$
(23)

The true maximum value will be the maximum value among the set of all the events that have occurred in the according time interval. We can obtain the probability density function of the true maximum value, conditioning on the number of total events, K, as follows:

$$f_{true\max} = \sum_{N=1}^{\infty} f_{\max(X)|N} p_N(K)$$
$$= \sum_{N=1}^{\infty} N(F_X(x))^{N-1} f_X(x)\frac{e^{-\lambda T}(\lambda T)^N}{N!}.$$
$$= \lambda T f_X(x) e^{-\lambda T(1-F_X(x))}$$
(24)

We can use this to calculate the expected value of the true maximum, $m_{true\,max}$

$$m_{true\max} = E[MAX(S)] = \int f_{true\max} x dx.$$
(25)

Similarly, the probability density function of the estimated maximum value, can be obtained as the density function of the maximum value of the set of included values, conditioning on the number of the received events, $|L|$:

$$f_{est\max} = \sum_{N=1}^{\infty} f_{\max(X)|N} \, p_N(|L|)$$

$$= \sum_{N=1}^{\infty} [N(F_X(x))^{N-1} f_X(x) e^{-\lambda(T-e^{-\mu h}\frac{(1-e^{-\mu T})}{\mu})}$$

$$\cdot \frac{(\lambda(T-e^{-\mu h}\frac{(1-e^{-\mu T})}{\mu}))^N}{N!}]$$

$$= \lambda(T-e^{-\mu h}\frac{(1-e^{-\mu T})}{\mu}) f_X(x)$$

$$\cdot \exp[-\lambda(1-F_X(x))(T-e^{-\mu h}\frac{(1-e^{-\mu T})}{\mu})]$$

(26)

and calculate the respective expected value

$$m_{est\max} = E[Max(L)] = \int f_{est\max} x \, dx. \qquad (27)$$

The expected value of the error in the maximum value estimation will then be:

$$E[ERR_{MAX}] = m_{true\max} - m_{est\max}. \qquad (28)$$

For example, if event values are uniformly distributed in an interval $[0,a]$, then it can be shown that

$$m_{true\max} = a - \frac{a}{\lambda T}(1-e^{-\lambda T}),$$

$$m_{est\max} = a - \frac{a\left[1-\exp\left(-\lambda\left(T-\frac{e^{-\mu h}(1-e^{-\mu T})}{\mu}\right)\right)\right]}{\lambda\left(T-\frac{e^{-\mu h}(1-e^{-\mu T})}{\mu}\right)},$$

which give a closed-form solution to (28).

Since in general absolute error and squared error do not yield closed-form solutions, we turn to a simpler error measure: the indicator error, whose mean gives the probability of getting a wrong estimate. In particular, if all event intensity values are i.i.d., then the arrival timestamp of event with the maximum value is uniformly distributed within each period. Therefore, the probability of missing this event is the same as the marginal probability of missing any other event, i.e.

$$P_e = e^{-\mu h} \frac{(1-e^{-\mu T})}{\mu T}. \qquad (29)$$

In general, the absolute and squared error analysis for exemplary aggregation functions, such as min/max and p-percentile, do not give closed-form solutions. However, the result (29) holds for any exemplary aggregation function that is solely based on event values, e.g., percentile.

IV. SIMULATIONS

To verify the analysis, we simulate the proposed aggregation functions under various settings. Specifically, we evaluate the quality of aggregation using the normalized mean absolute error, defined as the ratio of mean absolute error and expected true value, and plot the error versus network parameters (e.g.,

communication delay and rate of event streams) as well as design parameters (e.g., aggregation period and waiting time). Moreover, to evaluate the influence of event intensity, we simulate two intensities---uniform and lognormal---with the same mean and variance[1]. The normalized squared error shows the same trend and is therefore omitted due to space limit.

We first plot the normalized error as a function of the mean communication delay ($1/\mu$), as shown in Figure 3. From the Figure, we observe that the error is monotonically increasing with communication delay, as expected. Moreover, count and sum have identical error (because after normalization, the errors for count and sum are both equal to $e^{-\mu h}(1-e^{-\mu T})/\mu T$, which is the probability for an event to miss the deadline), which is much larger than the errors for average and max functions.
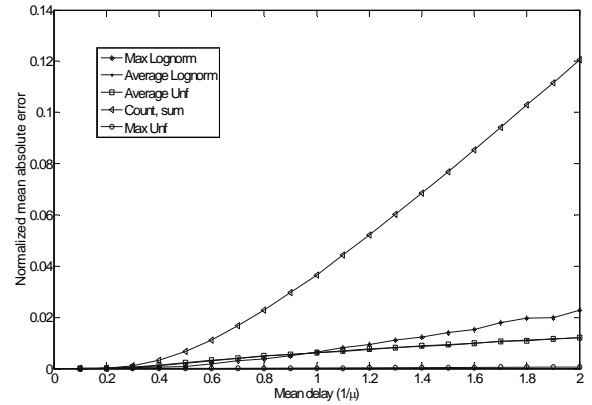

Figure 3 Normalized mean absolute error vs. Mean delay (λ_{tot}=20, T=10, h=1, 20000 iterations)

Furthermore, the event intensity does not affect the error except for max function, where the error for lognormal distribution is much larger than that for uniform distribution. This is because all the other functions only rely on the first moment of the intensity, whereas max relies on outliers and is therefore sensitive to how spread out the intensity is. These observations also hold for the other graphs (Figures 4 and 5). The simulations are consistent with the analysis whenever closed-form solutions are available[2].

Next, in Figure 4, we plot the error versus mean latency to further evaluate the trade-off between timeliness and accuracy. Timeliness is expressed in terms of mean latency, which is defined as the expected value of the time between the generation of an event and the time when it is included in the aggregation[3], given by $l=T/2+h$. Accordingly, we can adjust the mean latency by either varying T or varying h. As expected, all the errors decay as latency increases. Comparing the two

---

[1] We use a uniform distribution on [0,10] and a lognormal distribution derived by a normal distribution with mean m_N= 1.4656 and variance σ^2_N=0.2877. Both distributions have a mean equal to m_X= 5 and variance σ^2_X=8.33.

[2] For the average function, the upper bound is much larger than simulation results and is omitted for clarity.

[3] Within this context (events are time stamped and dropped if they miss the deadline), latency is only defined for the included events.

scenarios, however, we see that there exists a threshold latency, below which varying *T* gives better tradeoff and the opposite holds otherwise. This is because the latency increases twice slower with *T*, but the error decays much faster with *h*. From a design perspective, this means that given overall latency constraint, there is an optimal pair of aggregation period and waiting time that minimizes the error.
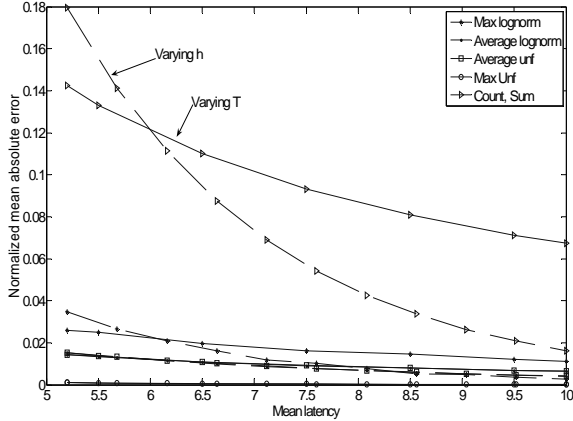

Figure 4 Normalized mean absolute error vs. Mean latency: varying T (solid line) or h (dashed line) (λtot=20, 1/μ=2, 20000 iterations. Solid line: T varies from 8.4 to 18, h=1; dashed line: h varies from 0.2 to 5, T=10)

Lastly, in Figure 5, we present the effect of the total event generation rate (λ_{tot}) on accuracy. As expected, λ_{tot} has the least impact on the accuracy compared with the other parameters. The errors for count and sum are not functions of λ_{tot} as shown before, and those for average and max only decay slightly.
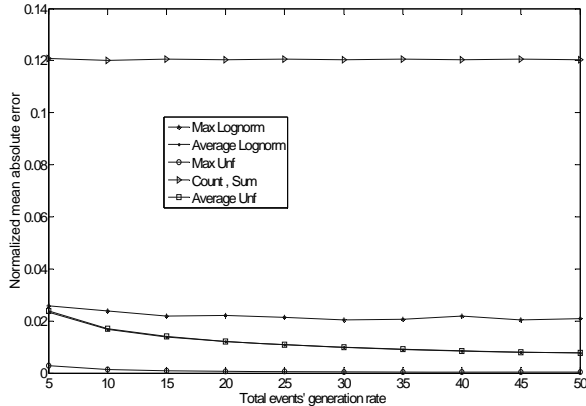

Figure 5 Normalized mean absolute error vs. Event generation rate λtot (h=1, T=10, 1/μ=2, 20000 iterations)

To summarize the simulation results, count and sum have the lowest accuracy, and the relative accuracy is (almost) independent of properties of event streams including event intensity and rate (except for max function, which is highly dependent on the intensity). For design purpose, we can achieve the desirable accuracy-timeliness tradeoff by properly choosing aggregation period and waiting time.

V.   RELATED WORK

Data aggregation has been a topic of intense study in the area of sensor networks, due to its potential for reducing the number of transmissions required to effectively monitor various physical phenomena, by providing only aggregate statistics on the sampled data. This, in turn, results in significant energy savings, which extends the lifetime of the resource-constraint sensor nodes. [6][8][20] study aggregation with respect to the specific underlying sensor network topology (e.g. tree-based, multiple path, or hybrid) that is used in order to transfer the partial aggregates. In contrast, our work is independent of the network topology and considers only the timing of the event generation process and the distribution of values of the events. It also provides a theoretical analysis of the error in the computation of aggregation functions and complements experimental evaluations such as [9] and [15].

Quality of Data (QoD) and Quality of Information (QoI) with respect to data aggregation are discussed in [7][10][11], primarily from the experimental perspective, while [11] gives an overall framework for QoI focusing on attributes such as completeness and confidence. This paper provides a theoretical view along the latency-accuracy axes using point processes and compliments the above work.

Efficient algorithms for computing aggregation functions have been investigated for network monitoring purposes using a hierarchical model [14][17] for the computation, a flat (2-tier) model [13], as well as fully decentralized approaches [12] based on gossiping. The emphasis is placed either on the maintenance of the spanning tree of the management overlay that computes the aggregate [14], the trade-offs between accuracy and communication cost [13][17][19], or the convergence of the algorithms to the aggregation values and resilience to nodes failures [12]. In contrast, in this work, we study the distribution of event arrival times (timeliness) at the processing center and their effect on accuracy, not the communication costs incurred in the fusion network or its topological assumptions.

In [16], the problem of aggregation is considered in the context of peer-to-peer network, in which nodes can dynamically join and leave during the computation of the aggregate. The focus is on the conditions that need to be satisfied in order to obtain valid results for the aggregation functions. Finally, recent work in [18] studies the problem of computing aggregate functions such as sum, count, mean and median on data streams with unreliable and noisy data and proposes algorithms for continuous computation of estimate of aggregates. Our work explores periodic computation of aggregates, as the monitored events accumulate over a time period specified as a design parameter of the monitoring system.

VI.   CONCLUSION

Data aggregation of network measurements is an important function of network monitoring systems, as it summarizes performance indicators of the network into metrics that are

easily assimilated by network administrators and are further used as a basis for SLAs with customers. For this reason, the ability to characterize the accuracy of the aggregate computation when measured samples are missing or delayed becomes of paramount importance, especially in large, distributed network monitoring infrastructures where the collection and processing of measurements are performed at disparate sites.

This paper provides an analytical study of the error that is introduced into the calculation of typical aggregation functions (sum, count, average and min/max), as they're computed periodically using measurements that might arrive late for computation at the network management center for processing. A model based on point processes is used for analyzing the arrival process of monitored events, and, based on this model, closed-form solutions and error bounds are provided for calculating the expected value and variance of the error. Simulation results verify our analysis and provide further insights on the accuracy-timeliness tradeoffs as we change network variables such as rate of events and design parameters such as aggregation period.

APPENDIX

I. Distribution of missed events

Let X be the number of events generated within the time interval $[(k-1)T, kT)$ and M the number of events that arrive after the aggregation time, and are therefore not included in the aggregation process (missed events). By assumption the delays of the events, until they reach the aggregation point, follow an exponential distribution, D with mean $1/\mu$. We initially condition on $X=n$; let $T_1, T_2... T_n$ be the generation times of the n events occurred (obviously, $T_i \in [(k-1)T, kT))$. We define the following indicator function:

$$miss(T_i + D_i) = \begin{cases} 1 & if \quad T_i + D_i > kT + h \\ 0 & if \quad T_i + D_i \le kT + h \end{cases}$$

Which is the indicator that the event will be missed. Then,

$$p = \Pr\{miss(T_i + D_i) = 1\} = \Pr\{T_i + D_i > kT + h\}$$

$$= \sum_{n=0}^{\infty} \frac{1}{T} \int_{(k-1)T}^{kT} \Pr\{Di > kT + h - u\} du \Pr\{X = n\}$$

$$= \frac{1}{T} \int_{(k-1)T}^{kT} e^{-\mu(kT+h-u)} du = \frac{1}{\mu T} e^{-\mu h}(1 - e^{-\mu T}).$$

Since the probability p that an event is missed is independent and identically distributed for all events, by the thinning argument of the Poisson process, the number of occurrence of such missing events within $[(k-1)T, kT)$ also follows a Poisson distribution with mean

$$p\lambda T = \lambda \frac{1}{T\mu} e^{-\mu h}(1 - e^{-\mu T})\mathrm{T} = \lambda e^{-\mu h}\frac{(1-e^{-\mu T})}{\mu} \ .$$

REFERENCES

[1] D. Verma, B. Ko, P. Zerfos, K. Lee, M. Duggan, K. Stewart, B. Rivera and A. Swami, "Understanding the Quality of Management in Computer Networks", under submission to ACITA 2008.

[2] H.M.Taylor and S.Karlin, *An Introduction to Stochastic Modeling*, Rev. ed. ed San Diego ; London : Academic Press, 1994.

[3] Leo A.Goodman, "On the Exact Variance of Products," *Journal of the American Statistical Association*, vol. 55, no. 292, pp. 708-713, Dec.1960.

[4] AT&T Managed Internet Service (MIS), http://new.serviceguide.att.com/mis.htm, 2008

[5] NTT Communications Global IP Network Service Level Agreement (SLA) http://www.us.ntt.net/support/sla/network/, 2008

[6] A. Deligiannakis, Y. Kotidis and N. Roussopoulos, "Hierarchical In-Network Data Aggregation with Quality Guarantees", in Proceedings of EDTBT 2004.

[7] A. Klein, "Incorporating Quality Aspects in Sensor Data Streams", in Proceedings of ACM PIKM 2007, Lisboa, Portugal.

[8] A. Manjhi, S. Nath, P. Gibbons, "Tributaries and Deltas: Efficient and Robust Aggregation in Sensor Network Streams", in Proceedings of SIGMOD 2005.

[9] J. Zhao, R. Govindan and D. Estrin, "Computing Aggregates for Monitoring Wireless Sensor Networks", in Proceedings of SPSN 2003.

[10] Biswas, F. Naumann and Q. Qiu, "Assessing the Completeness of Sensor Data", in Proceedings of DASFAA 2006.

[11] S. Zahedi, C. Bisdikian, "A Framework for QoI-Inspired Analysis for Sensor Network Deployment Planning", in Proc. PWSN 2007

[12] F. Wuhib, M. Dam, R. Stadler and A. Clemm, "Robust Monitoring of Network-wide Aggregates through Gossiping", in Proceedings of 10th IFIP/IEEE International Symposium on Integrated Management, 2007.

[13] C. Olston, J. Jiang and J. Widom, "Adaptive Filters for Continuous Queries over Distributed Data Streams", in Proceedings of SIGMOD 2003.

[14] A. G. Prieto and R. Stadler, "Adaptive Distributed Monitoring with Accuracy Objectives", in Proceedings of SIGCOMM Internet NM Workshops 2006.

[15] M. Sharaf, J. Beaver, A. Labrinidis, P. Chrysanthis"Balancing Energy Efficiency and Quality of Aggregate Data in Sensor Networks", in VLDB Journal 2004

[16] M. Bawa, H-G Molina, A. Gionis, R. Motwani, "Estimating Aggregates on a Peer-to-Peer Network", Stanford Technical Report 2003.

[17] N. Jain, D. Kit, P. Mahajan, P. Yalagandula, M. Dahlin and Y. Zhang, "STAR: Self-Tuning Aggregation for Scalable Monitoring", in Proc. of VLDB 2007

[18] T. Jayram, A. McGregor, S. Muthukrishan and Erik Vee, "Estimating Statistical Aggregates on Probabilistic Data Streams", in Proc. PODS 2007

[19] G. Cormode et. al. "Holistic Aggregates in a Networked World: Distributed Tracking of Approximate Quantiles", in Proc. SIGMOD 2005

[20] S. Madden, M. Franklin, J. Hellerstein and W. Hong, "TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks", in Proc. OSDI 2002