RC24560 (W0805-092) May 13, 2008
Computer Science

IBM Research Report

Understanding the Quality of Management in Computer Networks

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

Matthew Duggan, Kristian Stewart
IBM United Kingdom
South Bank
London, SE1 9PZ, UK

Brian Rivera, Ananthram Swami
Army Research Laboratory
Adelphi, MD 20783, USA

Understanding the Quality of Management in Computer Networks

Dinesh Verma†, Bong-Jun Ko†, Petros Zerfos†, Kang-Won Lee†, Matthew Duggan‡, Kristian Stewart‡, Brian Rivera§, Ananthram Swami§ [1]

†*IBM T. J. Watson Research Center, Hawthorne, NY 10954, U.S.A.*
‡*IBM United Kingdom, South Bank London, SE1 9PZ, UK.*
§*Army Research Laboratory, Adelphi, MD 20783, U.S.A.*

***Abstract*—The vitality and utility of a network is affected significantly by the network management system which is used to administer the network. However, in the state of the art, there are no good models that can characterize how good a network management system is. In this paper, we introduce the concept of Quality of Management (QoM), provide a mathematical formulation based on stochastic processes that can be used to model a network management system and define QoM metrics based on this formulation. A formal analysis of the proposed framework along various metrics is provided, along with examples of its application to several network management paradigms.**

***Index Terms*—quality of management, network management, discrete event model, aggregation functions**

I. Introduction

Network management is a key component of overall operations of any computer communications network. The choice of network management system can significantly affect the performance and utility of the network and impact its operating costs. It can also be a decisive element in the formulation of service level agreements that are established with its users. Because of the significance of management operations, it is important to ask how well a management system is meeting its objectives. However, the techniques used to determine whether a network management system is performing at a satisfactory level or not tend to be ad-hoc at present. Given the importance and vital role of network management in the operation of networks, there is a need for a formal framework and mathematical model that can characterize how well a network management system is working, or how good the *Quality of Management* is.

The Quality of Management (QoM) of a network management system characterizes how well the network management system achieves its purpose. Given any two network management systems, QoM should be able to describe which of the network management systems is better for a specific network management task. Frameworks for characterizing data quality have been studied in the areas of sensor networks [1][10][13], web searches [2], data stream processing systems [15], voice [3] and video [4] quality, and network games [5]. Also trade-offs among various dimensions of quality such as accuracy of information, completeness and transmission overhead have been investigated in the above areas (e.g. [11], [12]), while aspects over quality of monitoring primarily focusing on efficiency and accuracy have been the focus of recent studies in the network management field [20][21][22]. Complementary to the above efforts, the goal of this work is to propose a unified quantitative framework for assessing QoM for network management systems along several dimensions, without being bound to any particular protocol or system for network management.

Towards developing a comprehensive framework for QoM, this work makes the following contributions:

- A mathematical model based on stochastic processes for characterizing the management function of a network. The proposed model is generic enough to capture different types of management system functions, yet amenable to analysis of the key Quality of Management metrics.
- QoM metrics analyzed within the proposed stochastic framework with an emphasis on the effects of transmission latency on the accuracy of monitoring information that is obtained by the management system. We consider two types of management systems: (a) the transparent system where monitored events are immediately reported without processing and (b) the aggregation system, in which monitored events are periodically reported after aggregation.

The reminder of this paper is structured as follows: Section 2 provides a review of network management systems. Section 3 presents a simple model for the network management function and definitions. Section 4 presents a mathematical analysis of QoM metrics. Section 5 presents related work, and finally Section 6 concludes the paper with potential future research directions.

[1] Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defense and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defense or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

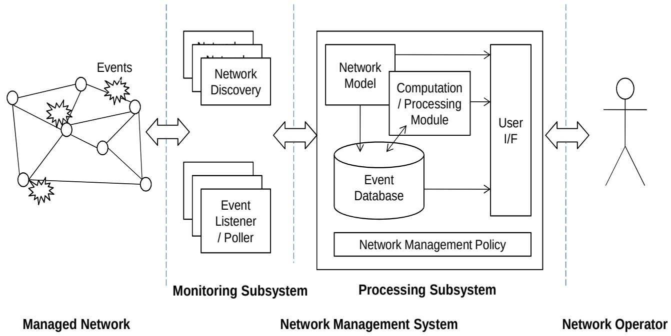

Figure 1 Typical Components of Network Management Systems

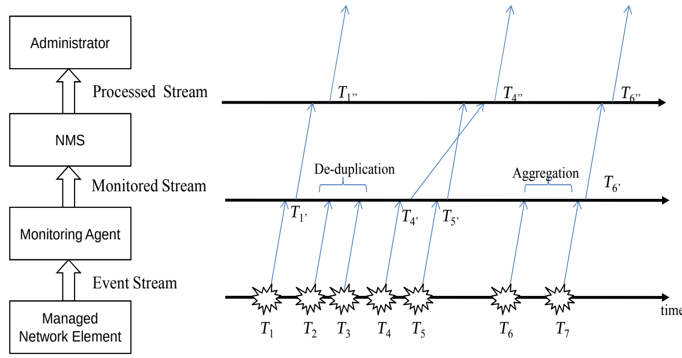

Figure 2 Information Processing Sequence in a Network Management System

II. Network Management System

To ensure continuous and efficient operation of a network, the operator uses a network management system to discover elements in the managed network, identify any situations that may need attention, and then take any requisite corrective actions. Three distinct entities can be readily identified in this process, namely the managed network itself, the network management system, and the network operator, as shown in Figure 1.

Managed networks are subject to a variety of unpredictable situations that may require the attention of a network operator, such as hardware and software failures, misconfiguration of devices or applications, as well as problems caused by third-party networks that are connected to the managed network. The network operator's management policy may dictate that information about some of these situations is either sent to or collected by the network management system. This information may be manifested as *events*, which typically have a number of characteristics including: (a) a message describing a situation, (b) an indication of the managed resources affected by the situation, (c) an indication of the originator of the event, and (d) an indication of the time the situation arose.

The network management system may receive events at various times in a number of diverse formats. Common types of events include SNMP traps, syslog messages and events originating from periodic polling of IP addresses and SNMP MIB variables. Events are typically subject to some degree of preprocessing so that they are consumable by other processes and/or the network operator for further analysis. For the purposes of this paper, we consider the network operator to be the consumer of events processed by the network management system.

Depending on the management policy, some events may be subject to additional processing. The system may decide to: (a) discard an event; (b) suppress reporting of duplicate events and instead increment a counter associated to the existing event[2]; (c) archive events for historical analysis and auditing; (d) analyze and correlate events in the context of a constructed network model so as to identify which events are considered 'cause' and which are 'symptoms'; (e) enrich events with additional information, such as customer contact information; (f) inject synthetic events back into the event stream[3]; and (g) send events to other network management systems.

After processing, events are typically presented to an administrator. Common ways of providing an administrator with event information includes the use of graphical user interfaces, SMS messages and e-mail. The administrator then may take some corrective action depending on the situation described by an event.

III. Modeling the Management Function

In order to understand QoM, we need to define a model of the network management function, which is generic enough to abstract from the specifics of the underlying implementations, yet representative of real-world deployed network management systems for the analysis to give useful insights. In this section, we present a stream model for network management systems and define the QoM metrics that will be used for the analysis in a later section.

A. Stream Model for Network Management

Following the aforementioned description of the typical network management system, two flows of information can be readily identified in the management environment: one between the managed network elements to the network management system, and another one between the network management system and the operator. In large scale network management deployments, the network management system usually consists of distributed network monitoring agents that are close to the managed entities and report the results to a centralized network management system (NMS). Once an event in the network is detected by the monitoring agents, simple processing is handled first by the agents, and the event information is typically sent to the NMS for further calculation of user-defined metrics. Then, the results of the analysis are presented to the network administrator. Thus, the sequence of information processing in an NMS can be modeled using a

[2] This operation is commonly called *de-duplication* in the network management context.

[3] For example, a link-flapping event may be deduced from repeated occurrences of link-down/link-up events and/or ping failure/restoration events.

three-stream model, as shown in Fig. 2. In a centralized network management system where all the monitoring and polling operation is performed by a single NMS, the interaction is usually direct from the managed network to NMS.

When an external event such as link failure happens in the system, it has an impact on the operation of the network. The occurrence of these external events is represented by the *Event Stream* in the Figure 2, with occurrence times T1 … T7. Some of these event instances are collected and pre-processed by the monitoring agents, and sent to the network management. This is called the *Monitored Stream*. In the same figure, we present the case when some events are suppressed (e.g. events occurred at T2, T3) by the monitoring agent. These monitored events are sent to the NMS for further processing (e.g. event correlation) and can be optionally stored in a database for later use. The information that is generated as output from the NMS is called the *Processed Stream*. Since the monitoring agents may be remotely located, the NMS may receive the monitored data out of sequence depending on the network condition as shown in the case of T4' and T5'. The information from the processed stream is further reported to the administrator using some means as previously described. This is represented as the *Administrator View* in the figure. We note that the generic processing model shown in Figure 2 is applicable to most network management systems.

Quality of management should be analyzed by comparing the processed stream with the event stream, the latter representing the actual state of the network and being susceptible to measurement errors induced by the monitoring agents, as well as event transmission losses due to network instabilities. To simplify though our analysis in the next section, we assume perfect monitoring agents for the NMS, which introduce no errors in the measurements or the transmission of events, and study the event processing that takes place between the event stream and the processed stream under this assumption.

B. *Formal Definition of a Stream Model*

Formally, we define the stream model as follows. Let $E = \{e_1, e_2, \ldots e_k\}$ be an enumerable set of events that are of interest in a management context. The event stream *ES* is a stochastic process defined over the set *E*. Examples of events include:

- The link between address 9.2.10.1 and 9.2.10.25 fails.
- The number of dropped packets at node with address 9.2.10.20 exceeds 10% of all packets.

Each event is assumed to be a discrete entity. In a finite network, the number of distinct events of interest in the entire system can be assumed to be enumerable and finite without loss of generality. In the event stream *ES*, each event is associated with a time at which the event occurs.

Similarly, let $M = \{ m_1, m_2, \ldots m_{k'} \}$ be an enumerable set of monitored data. A monitored data is a unit of information *detected* by the network monitoring agent. The monitored stream *MS* is a stochastic process defined over set *M*. We assume that each monitored data is carried in a single logical message (the message may be transmitted on multiple network packets if needed), and contains some information about one or more events in the network. Examples of monitored data may be:

- A polling response stating that node with address 9.2.10.11 is up.
- A trap from node 9.2.10.21 indicating that its neighbor 9.2.10.22 is no longer reachable.

Let $P = \{ s_1, s_2, \ldots s_k \ldots\}$ be an enumerable set of records in the processed data stream received by the network management system. The processed stream *PS* is a stochastic process defined over the elements in set *P*. Similarly to the monitored data stream, individual processed stream data is carried in a single logical message (the message may be transmitted on multiple network packets if needed), and contains information about the events in the network that has been derived from processed and correlation of monitored data. For example, processed stream data may be:

- Route flapping happened between nodes 9.2.10.21 and 9.2.10.22.
- Node 9.2.10.12 could not be reached for the last five minutes.

C. *QoM Metrics*

The three streams characterizing the management system can be analyzed along different metrics. We consider three primary measures for evaluating the quality of management: (a) *latency*, (b) *accuracy*, and (c) *efficiency*.

Latency: This metric denotes the timeliness of a network monitoring function. Latency *L* is defined as the expected time between the moment an event appears in the event stream and when the indication of that event appears in the processed stream.

Accuracy: Accuracy characterizes the ability of the management process to capture the events that happen in the event stream. It is defined as the fraction of the time for which the status of the network as measured by the processed stream is consistent with the actual status of the network as represented by the event stream. If we had an oracle that could examine the status of both streams, this fraction could be accurately determined. The accuracy measure defined in this way will be dependent upon the latency of the network, as well as the probability of the loss of processed stream data in the network and errors in the measurement process of the monitoring agents.

Efficiency: Conceptually, efficiency can be measured by the volume of data that is associated with an event in the event stream. However, events in the event stream are not associated with any volume of data, as they are considered dimensionless. This implies that efficiency cannot be obtained by comparing volume of data at the processed stream with the respective one at the event stream. Therefore, alternative practical metrics of efficiency are used, such as the average frequency and average data volume of the processed data streams to those of the monitored stream, respectively, for an event in the event stream.

We should note that the filtering and aggregation of events in the processing system affect the accuracy and efficiency of a

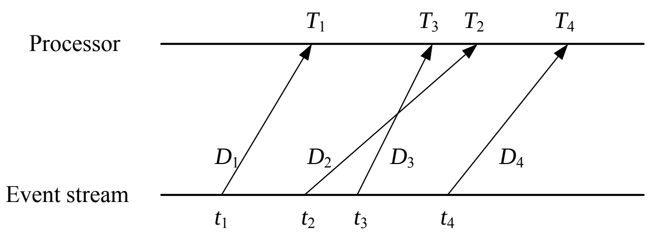

Figure 3 Event timing model

monitoring system. Moreover, information associated with events might have to be incorporated, for example by adding metadata to an event for enriching its description (an operation known as "enriching"). This means that it might not be sufficient to represent the event streams at the monitoring and processed streams as point processes that follow the "birth" of the events, but information on the data volume associated with the events will also need to be included. In this paper, we focus on the timing behavior of the events' delivery over the network, and provide more enriched analysis of the impact of aggregation function on the accuracy in a companion paper [26].

A management operation may have very different characteristics for the three dimensions. For example, it may exhibit higher accuracy but a higher volume than another management or data collection operations. The relative merit of the three different axes of the Quality of Management would depend on which of the different criteria are considered most important in a particular management environment.

IV. QoM Analysis

In this section, we investigate generic properties of network management systems using formal stochastic analysis techniques under our QoM model. We consider the following nominal network management model for the analysis:

- A distributed network monitoring agent is deployed in each managed network element, configured to continuously monitor the events in the *event stream*. We assume "transparent" network monitors, meaning that all events of interests to the network administrator are captured with no delay by the monitoring agents (or with delay small enough to neglect in the overall latency within NMS).
- Upon capturing a network event, each monitoring agent immediately sends the monitored management data to a centralized processing system. We assume the management data is reliably delivered with some networking delay from each monitoring agent to the processing system without loss. The monitored data sent by all monitoring agents constitutes the *monitored stream* in our stream model.
- The processing system in turn stores and processes these monitored data sent by the monitoring agents across the network in some manner (we will describe the processing methods shortly) and reports the processed data to the administrator. This report to the administrator constitutes the *processed stream*.

Note that, although we assume reliable data delivery between the monitoring agents and the processing center, we do take into account late delivery of the data from monitoring agents to the processing center. We leave the analysis of a more comprehensive model that incorporates the monitoring and networking loss as a future research item.

We consider the following timing model of events and network delay (see Figure 3). Let t_i denote the time that i-th event occurs in the event stream ($t_i < t_{i+1}$ for all $i = 0, 1, \ldots$), T_i the time that i-th event arrives at the processing system, and D_i the difference between T_i and t_i, i.e., $D_i = T_i - t_i$. Then we assume:

- Events in event stream occur according to a homogeneous Poisson process with arrival rate λ.
- The network delay D_i is i.i.d. random variable with exponential distribution of mean $1/\mu$, i.e., $\Pr\{D_i > t\} = e^{-\mu t}$.

Note that, since we assume transparent monitors (i.e., no loss and zero delay) the monitored stream is essentially the same as the event stream, and therefore omitted in Figure 3. Note also that, although our results are derived based on the assumption of exponential network, analysis for other delay distributions is also possible. For instance, one can analyze the accuracy metric using an M/G/∞ rather than the M/M/∞ queuing model.

We analyze the QoM metrics for two representative management data processing methods, namely the transparent system and the aggregation system. In the *transparent system*, the processor immediately reports each event that it receives to the administrator. This type of processing is most applicable when the goal of the network monitoring is to follow the real-time activities (for instance, link failure, threshold crossing alarm, etc.) in the event stream and all necessary data aggregation takes place in the lower level, for example, by the network monitoring agents residing in the managed entities.

The *aggregation system* periodically aggregates the events sent by the monitors, and reports (and stores) the aggregated events to the administrator with aggregation period T. If there is no event received during an aggregation period, no aggregated event is generated during that period. In this sense, T=0 is a special case that represents a transparent system. This type of processing method is most appropriate when the managed entities produce raw data and the management system is responsible for the aggregation and further processing before reporting to the administrative stream.

A. QoM Analysis of Transparent System

In the transparent stream, since there is a one-to-one mapping between events in the event stream (ES) and the data in the processed stream (PS), the efficiency, measured by the ratio of frequency of events in ES to that of the data generation in PS is 1. Also, since there is no additional delay within the processor, the latency, L_i, of an event i in ES is exactly the random delay, D_i. Hence the latency distribution is also exponentially distributed, i.e., $\Pr\{L_i > t\} = \Pr\{D_i > t\} = e^{-\mu t}$, and $E[L_i] = E[D_i] = 1/\mu$.

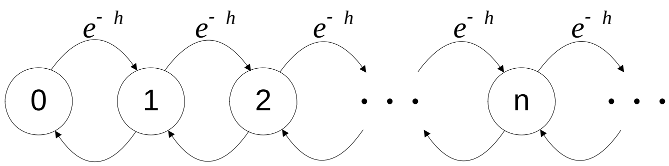

Figure 4 M/M/∞ process model: Number of out-of-sync events

We define the accuracy of the transparent management system as *the fraction of time that the network administrator is in-sync with the event stream*, where the *synchrony* is defined in the following way: We say that NMS is in an *out-of-sync* state when there is any event that arrives at the processed stream later than a constant time h from the moment the event is generated in the event stream (Otherwise NMS is in the *in-sync* state). This constant time h is a design parameter, typically introduced to enable the NMS to account for the network delay, and is used as a maximum allowable amount of time that the NMS can wait before presenting a delivered event as in-sync event.

Suppose the NMS is designed in such a way that it holds the events received before time $t+h$ (t is the event occurrence time) until $t+h$ before presenting the events to the administrator (e.g., for providing a synchronized real-time view of the events). The average latency in this case is then $E[L_i] = h \Pr\{D_i > t\} + (h+1/\mu)\Pr\{D_i > t\} = h+1/\mu$.

To analyze the fraction of time that the NMS is in the in-sync state, we define a continuous-time stochastic process, $Q(t)$, that represents the number of outstanding out-of-sync events as follows:

- Initially, $Q(t) = 0$ at $t = 0$.
- Every time an event i that is generated at time t_i arrives at the processor at some time $T_i > t_i+h$, $Q(t)$ is increased by 1 at t_i+h, and decreased by 1 at T_i.

We now show that the stochastic process $Q(t)$ can be represented by a M/M/∞ process with some arrival rate λ_q and some departure rate μ_q, and the accuracy of the management system can be calculated by finding the limiting distribution $\lim_{t\to\infty}\Pr\{Q(t) = 0\}$.

First, let us define a new arrival process $\{v_i\}$ as the time instances that shift the events arrival times by h, i.e., $v_i = t_i+h$. Since shifting the arrival times of a homogeneous Poisson process is also Poisson with the same rate [25, pp. 318-319], the arrivals of $\{v_i\}$ is also Poisson with the same rate λ.

Furthermore, let us define random variables $\{S_i\}$ such that S_i =1 if the processor has not received the event i at time s_i (otherwise S_i =0). We further define $\{V_i\}$ as the subset of $\{v_i\}$ where S_i =1, i.e., the moments that the processor stream starts to be out-of-sync with the event i.

Now consider an event i that occurs at times t_i. Then probability that this event is delivered at the processor later than t_i+h is

$$\Pr\{S_i = 1\} = \Pr\{T_i > t_i + h\} = \Pr\{D_i > h\} = e^{-\mu h},$$

from the exponential distribution of the network delay. Since the occurrence of an event's out-of-sync delivery is i.i.d., is a Bernoulli process. Therefore the arrival process $\{V_i\}$ is a

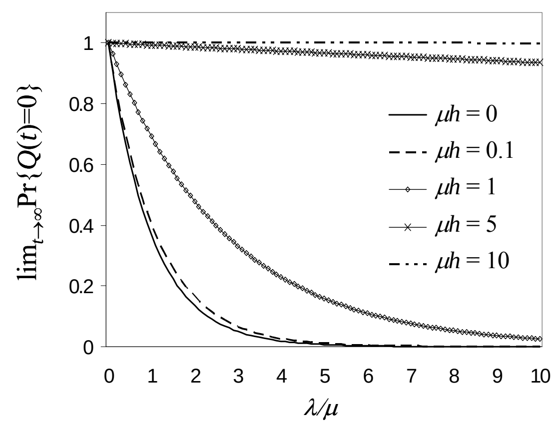

Figure 5 Accuracy as the steady-state probability of being in the in-sync state.

thinned Poisson process with arrival rate $\lambda \Pr\{S_i = 1\} = e^{-\mu h}\lambda$. Since $Q(t)$ is increased by 1 at each $t \in \{V_i\}$, the transition of $Q(t)$ from any k to $k+1$ is Markovian and occurs at rate $\lambda_q = e^{-\mu h}\lambda$.

By the memory-less property of the exponential distribution, it is straightforward to see the time until an out-of-sync event finally arrives at the processor is also exponentially distributed with mean $1/\mu$. More specifically, $\Pr\{D_i > (t+h)+s \mid D_i > t+h\} = \Pr\{D_i > s\} = e^{-\mu s}$. Therefore, the process $Q(t)$ is indeed a Markov process, and more precisely an M/M/∞ process with arrival rate $\lambda_q = e^{-\mu h}\lambda$ and departure rate $\mu_q=\mu$, whose state transition diagram is depicted in Figure 4.

The accuracy of the management system is therefore obtained by the standard limiting distribution of the M/M/∞ process, that is, [4]

$$\lim_{t\to\infty} \Pr\{Q(t) = 0\} = e^{-\lambda_q/\mu_q} = e^{-\exp(-\mu h)\left(\frac{\lambda}{\mu}\right)}. \qquad (1)$$

Note that the steady-state probability of being in the in-sync state goes to zero as the ratio λ/μ becomes large, i.e., as the mean arrival rate of events becomes large relative to the mean delay. Also the accuracy increases as μh increases, i.e., the waiting constant h is set relatively larger than the average latency $1/\mu$ (the penalty is increased latency of $h+1/\mu$)

Figure 5 shows the accuracy of the transparent system as a function of λ/μ for different values of μh. For instance, when $h=1/\mu$ and when this ratio is as large as 2, there is a 50% chance that the system view will be out-of-sync. Thus practical NMS systems must be designed to ensure the network delay $1/\mu$ relatively small to the mean inter-arrival time of the events $1/\lambda$.

If the network delay distribution is not exponential, it is a relatively straightforward exercise to verify that the service

[4] More generally, the limiting distribution of the number of outstanding out-of-sync events is $\lim_{t\to\infty} \Pr\{Q(t) = k\} = (\lambda_q/\mu_q)e^{-\lambda_q/\mu_q}/k!$. Understanding the temporal evolution of the statistics of the synchrony is also important and can be analyzed using our Markov model, for instance, how fast $Q(t)$ approaches the steady-state.

time of the out-of-sync events no longer enjoys the Markov property, while the inter-arrival time between out-of-sync events is still exponentially distributed. Hence the accuracy under non-exponential delay can be obtained by analyzing the resulting M/G/∞ model.

B. *QoM Analysis of Aggregation System*

Suppose the processor periodically aggregates at time $t = kT + h$ the events that occurred in the event stream within the time epoch $[(k\text{-}1)T,\ kT)$ at time $t = kT + h$ for some constant h and for $k = 1, 2, \ldots$. The additional time h is introduced as a "waiting constant" in order to allow the events occurring near the end of each epoch to arrive at the processor and be aggregated with other events that occurred in the same epoch. However, even with this additional waiting time, due to the randomness of delay, there can be some events that occur in $[(k\text{-}1)T,\ kT)$ but miss the deadline $t = kT + h$, in which case the late events will be aggregated at the earliest aggregation time after it finally arrives. We will investigate the impact of the choice of h on the latency and the accuracy later in this section.

With the aggregation taking place periodically, it is easy to see that the efficiency, defined as the ratio of the reporting frequency at processed stream to the frequency that events occur in the event stream, is $(1 - e^{-\lambda T})/(\lambda T)$.

The latency L_i of an event i in the event stream is the sum of the random network delay and the time between the arrival of i at the processor and the waiting interval until the next aggregation time. Without loss of generality, suppose an event i occurred at some time $t_i = t \in [0, T)$. Then the latency of this event is such that, If $t + D_i < T+h$, i.e., if this event arrives at the processor no later than the first aggregation time ($T+h$), then $L_i = T+h-t$. Note that, for *all* events that occur in , the first aggregation time is at $T+h$, not at h, i.e., even if an event arrives before time h, it will be aggregated at $T+h$, not at h. Otherwise, if the event i's arrival time at the processor ($t+D_i$) falls within $[kT+h,\ (k+1)T+h)$, then $L_i = kT+h - t$. for $k = 1, 2, \ldots$.

Since D_i is exponentially distributed with mean $1/\mu$, the probability that event i is aggregated at $T+h$ is $\Pr\{t + D_i < T+h\} = \Pr\{D_i < T+h - t\} = 1 - e^{-\mu(T+h-t)}$, and the probability that it will be aggregated at later aggregation time $kT+h$ (k=1, 2, …) is $\Pr\{kT+h < t + D_i < (k+1)T+h\} = e^{-\mu(kT+h-t)} - e^{-\mu((k+1)T+h-t)} = e^{\mu(t-h)}e^{-k\mu T}(1- e^{-\mu T})$ for $k = 1, 2, \ldots$. Therefore, the average latency $E_t[L_i|t]$ given $T_i = t \in [0, T)$ is

$$\begin{aligned} E[L_i \mid t] &= (T+h-t)\Pr\{D_i + t < T+h\} + \sum_{k=1}^{\infty}((k+1)T+h-t)\Pr\{kT+h < D_i + t < (k+1)T+h\} \\ &= (T+h-t)(1-e^{-\mu(T+h-t)}) + \sum_{k=1}^{\infty}((k+1)T+h-t)e^{-k\mu T}e^{\mu(t-h)}(1-e^{-\mu T}) \\ &= (T+h-t)(1-e^{-\mu(T+h-t)}) + (T+h-t)e^{\mu(t-h)}(1-e^{-\mu T})\sum_{k=1}^{\infty}e^{-k\mu T} + Te^{\mu(t-h)}(1-e^{-\mu T})\sum_{k=1}^{\infty}ke^{-k\mu T}. \end{aligned}$$

Using $\sum_{k=1}^{\infty} a^k = a/(1-a)$ and $\sum_{k=1}^{\infty} k a^k = a/(1-a)^2$ for $|a| < 1$,

$$\begin{aligned} E[L_i \mid t] &= (T+h-t)(1-e^{-\mu(T+h-t)}) + (T+h-t)e^{-\mu(T+h-t)} + T\frac{e^{-\mu(T+h-t)}}{1-e^{-\mu T}} \\ &= (T+h-t) + T\frac{e^{-\mu(T+h-t)}}{1-e^{-\mu T}}. \end{aligned}$$

Since the events in the event stream constitute a Poisson process, the distribution of events' arrival times in the event stream is uniformly distributed within $[kT,\ (k+1)T]$ [24, pp. 297-299]. Hence, the (unconditioned) mean latency $E[L_i]$ is

$$\begin{aligned} E[L_i] &= E_t[E[L_i \mid t]] = \frac{1}{T}\int_0^T \left((T+h-t) + T\frac{e^{-\mu(T+h-t)}}{1-e^{-\mu T}} \right) dt \\ &= T + h - \frac{T}{2} + \frac{e^{-\mu T}}{1-e^{-\mu T}}\int_0^T e^{\mu t} dt \\ &= \frac{T}{2} + \frac{1}{\mu}\frac{e^{-\mu(T+h)}(e^{\mu T}-1)}{1-e^{-\mu T}} = \frac{T}{2} + h + \frac{e^{-\mu h}}{\mu}. \end{aligned} \qquad (2)$$

An intuitive interpretation of Eq. (2) is that the latency $E[L_i]$ can be broken into three components: the first one is the delay until the mean aggregation time ($T/2$), the second is the additional delay due to the waiting constant (h), and the last is the additional delay affected by whether the event arrives after the nearest aggregation time ($e^{-\mu h}/\mu$).

The impact of the waiting constant (h) on the overall latency is such that, as h grows, $E[L_i]$ quickly approaches $T/2 + h$, which makes the latency grow linearly with h. On the other hand, if $\mu h << 1$, $E[L_i] \approx T/2 + 1/\mu$, leaving only the latency components due to the periodic aggregation. Also note that, as $T \to 0$ and $h \to 0$, the latency $E[L_i] \to 1/\mu$, which corresponds to the latency in the transparent system.

We now turn our attention to the accuracy. Due to the periodic aggregation, however, the accuracy analysis for the aggregation system based on the definition Section 4.1 does not enjoy the Markov property. Therefore, we instead use the following alternative definition for the accuracy of the system. We say the network management system is in in-sync state during the (entire) aggregation period $[(k\text{-}1)T,\ kT]$ if *all* events that occurred during this period arrive at the processor before time $t = kT+h$. Otherwise, the NMS's view is out-of-sync during that period. We define W_k as the random variable that takes 1 if the aggregation period $[(k\text{-}1)T,\ kT]$ is in-sync, and 0 otherwise; Then the accuracy of the system is defined as the probability $\Pr\{W_k = 1\}$, which is independent of k as all t_i's and D_i's are independent.

Suppose an event i occurs at some time $t_i = t \in [(k\text{-}1)T,\ kT]$. Then the conditional probability that its arrival time T_i at the processor is before $kT+h$ is $\Pr\{T_i < kT+h \mid t\} = \Pr\{D_i + t < kT+h\} = 1 - e^{-\mu(kT+h-t)}$.

Now suppose the number of events in $[(k\text{-}1)T,\ kT)$ is given by $N_k = n$. Since the arrival times of such n events are uniformly distributed within $[(k\text{-}1)T,\ kT)$,

$$\Pr\{T_i < T+h\} = \frac{1}{T}\int_{(k-1)T}^{kT}(1-e^{-\mu(kT+h-t)})dt = 1 - \frac{e^{-\mu h}(1-e^{-\mu T})}{\mu T},$$

and since all n events must arrive at the processor before $kT+h$ for the aggregation period W_k to be in-sync, our accuracy measure of an aggregate system is

$$\Pr\{W_k = 1 \mid N_k = n\} = \left(1 - \frac{e^{-\mu h}(1-e^{-\mu T})}{\mu T}\right)^n.$$

It follows from the Poisson arrival of the events that the unconditional probability of the NMS being in the in-sync state is

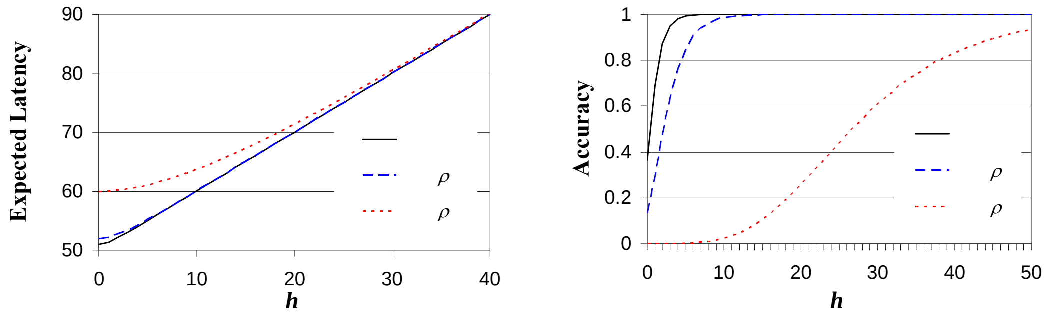

Figure 6 The expected latency and accuracy of the aggregation system as functions of waiting constant h (T=100, λ=1)

$$\Pr\{W_k = 1\} = \sum_{n=0}^{\infty}\left(1 - \frac{e^{-\mu h}(1-e^{-\mu T})}{\mu T}\right)^n \frac{(\lambda T)^n e^{-\lambda T}}{n!} \qquad (3)$$
$$= e^{-\exp(-\mu h)(1-\exp(-\mu T))\frac{\lambda}{\mu}}.$$

Similarly to the latency case, the impact of the waiting constant on the accuracy of the system is clear: if the waiting constant h is large, it becomes more likely that all events are delivered within the (extended) deadline provided by large h, and hence the management system is likely to maintain a in-sync view of the network. One can see this from $\Pr\{W_k = 1\}$ ≈ 1 when $\mu h >> 1$. Also, notice that the accuracy metric of the aggregation system approaches that of the transparent system when $\mu T >> 1$, i.e., $\Pr\{W_k = 1\} \approx e^{-\exp(-\mu h)\frac{\lambda}{\mu}}$ when $\mu T >> 1$.

We conclude the QoM analysis of aggregation systems with an interesting observation on the tradeoff between latency and accuracy, introduced by the waiting constant *h*. It can be seen from the analytical results in Eq. (2) and (3) that a large waiting constant *h* improves accuracy at the expense of increased latency, while a small *h* reduces latency but decreases accuracy.

In Figure 6, we plot numerical values of the average latency and accuracy respectively, as a function of *h* with *T*=100 and λ=1. We can observe that, with small waiting constant *h*, the overall latency is mainly determined by the average network delay—the latency is high with large network delay (small μ). However, the impact of network delay diminishes as *h* grows, and the latency is mostly determined by *h*. In the case of accuracy, increasing *h* has an effect of enhancing the accuracy, which converges to 1. But the rate at which the accuracy approaches 1 is largely dependent on the network delay ($1/\mu$): larger network delay causes lower accuracy. This tradeoff analysis facilitates the design of network management systems in selecting proper values of waiting time until aggregation and the aggregation period depending on the specific design goal (e.g., delay and accuracy constraints) and on the expected networking delay and event arrival characteristics.

V. Related Work

In [23], Chen and Liu provided a model for four network management approaches (client-server, hierarchical, weak and strong mobility) and evaluated them with respect to scalability and efficiency using simple static assumptions. Unlike [23], our proposed model enables us to analyze the dynamic behavior of QoM.

Data quality characterization and analysis have been the focus of study in the areas of sensor networks, data stream processing systems, and, in recent years, in network monitoring of distributed systems. In sensor networks, several efforts (e.g. [9][11]) explored the tradeoffs between energy consumption and precision of in-network computation of aggregate statistics such as AVG, MIN and COUNT. In [12], Zhao et al. proposed an architecture and protocols for digest diffusion of network aggregates for sensor network monitoring purposes. The focus is primarily on the topologies used for the in-network computation paths and how they affect approximation errors with respect to the above operators under packet loss and sensor node failures. In contrast, our work is focused on the effects of late delivery of events and the resulting discrepancy between the event and the administrator views. Recent work [10] has similar goal to ours: to provide a framework for assessing quality of information transferred and managed along dimensions such as accuracy, completeness (also discussed in [13]) and confidence. However, the focus is more on sensor imprecision and its effects on accuracy, rather than the effects of latency.

Since some of the most frequent operations in network monitoring are aggregations of the raw event data using various functions such as averaging, count, max, quality of network monitoring is conceptually similar to quality of data in stream processing systems. In this area, work on adaptive filters [17][19] explores the trade-off between precision and transmission overhead of data collected centrally from distributed sources. While latency and its effect on the correctness of source data are briefly discussed in [17], no quantitative framework is provided to characterize discrepancy of the monitored and the administrator view when the latency tolerance bounds are violated. [18] introduced the concept of fraction-based tolerance for adapting the width of the filter at the source depending on the how many false-positive or false-negative events (as opposed to an error-constraint in their value) the system is willing to tolerate. This bears similarity with our definition of the accuracy dimension in QoM, but this work does not discuss the effects of latency. [14] proposed three protocols that satisfy progressively less-strict levels of correctness and explore their behavior with respect to accuracy of computations, resilience to transmission delays and overhead. Although very similar to our work, [14] is concerned with the in-network processing of aggregate functions, and not as much with the discrepancy between various "views" of event data in the multi-level management hierarchy model.

Recently, accuracy of data in network monitoring applications has received renowned attention. Prieto et. al. in [20] formulate the task of computing aggregates from a hierarchical monitoring network graph with minimal overhead as a constraint optimization problem. However their work does not mention the effects of latency in propagating the partial computations to the root. Follow-up work in [16] evaluates the use of gossiping in the computation of aggregates and compares these approaches, without establishing though a generalized quality framework as is the purpose of this work. In [21] the trade-offs between monitoring overhead and accuracy are investigated from the sampling perspective, for dimensioning the monitoring infrastructure, but no comprehensive framework for the analysis is provided. Finally, [22] studies the problem of optimal sampling strategies of flow data whose traffic distributions are not known in advance. The model considered is similar to the events, monitored and processed streams of this paper. However, our work focuses on providing a framework for characterizing the discrepancy between the streams of monitored and stored data and not on the particular sampling strategy.

VI. Conclusion

Network management is fundamental in ensuring continuous and efficient operation of a communication network. Although various management architectures that have been proposed in the past a generic model for the analysis and comparison of these architectures has been missing. This paper introduced the concept of Quality of Management (QoM) and proposed a framework for analyzing it using the paradigm of event streams modeled as stochastic processes. Formal definitions of accuracy, efficiency and latency dimensions as metrics of quality in this framework are also provided. A study of QoM along the accuracy-latency dimensions is performed under this model, in which we show how the choice of aggregation waiting time affects the accuracy of the observed events at the monitoring level. Our current framework assumes that the operation of the management system does not impact the events in a network. In future work, we intend to extend the framework to account for the impact of management system operations on the network events and the effects of in-band management, and to further analyze the QoM with different stochastic characteristics of underlying stream and network model.

References

[1] S. Zahedi and C. Bisdikian, *A framework for QoI-inspired analysis for sensor network deployment planning*, 2nd Int'l Workshop on Performance Control in Wireless Sensor Networks (PWSN 2007), October 2007.

[2] C. Cappiello, C. Francalanci and B. Pernici, *Data Quality Assessment from the User's Perspective*, International Workshop on Information Quality in Information Systems, Paris, France, June 2004.

[3] ITU-T Recommendation P.862: Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs, February 2001. Available at URL http://www.itu.int/rec/T-REC-P.862-200102-I/en

[4] ITU Standard BT.500, Methodology for the subjective assessment of the quality of television pictures, June 2001, Available at URL http://www.itu.int/rec/R-REC-BT.500/en.

[5] C. Schaefer, T. Enderes, H. Ritter and M. Zitterbart, Subjectie Quality Assessment for MultiPlayer real-time games, Proceedings of the 1st workshop on Network and System Support for Games, Bruanschwig, Germany, April 2002, pp. 74-78.

[6] J A. Medina, A. Lakhina, I. Matta, and J. Byers. BRITE: An Approach to Universal Topology Generation. In Proceedings of the International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems- MASCOTS□'01, Cincinnati, Ohio, August 2001. http://www.cs.bu.edu/brite/.

[7] A Better Model for Generating Test Networks, M. Doar, IEEE Global Telecommunications Conference/GLOBECOM'96, London, November 1996.

[8] E. Zegura, K. Calvert and S. Bhattacharjee. "How to Model an Internetwork." Proceedings of IEEE Infocom '96, San Francisco, CA.

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

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

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

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

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

[14] M. Bawa, A. Gionis, H. G. Molina and R. Motwani, "The Price of Validity in Dynamic Networks", in Proceedings of SIGMOD 2004.

[15] U. Leser, F. Naumann, "Query Planning with Information Quality Bounds", in Proceedings of FQAS 2000.

[16] 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.

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

[18] R. Cheng, B. Kao, S. Prabhakar, A. Kwan and Y. Tu, "Adaptive Stream Filters for Entity-based Queries with Non-Value Tolerance", in Proceedings of VLDB 2005.

[19] A. Jain, E. Chang and Y.-F. Wang, "Adaptive Stream Resource Management Using Kalman Filters", in Proceedings of SIGMOD 2004.

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

[21] N. Duffield and C. Lund, "Predicting Resource Usage and Estimation Accuracy in an IP Flow Measurement Collection Infrastructure", in Proceedings of IMC 2003.

[22] N. Duffield, C. Lund and M. Thorup, "Learn More, Sample Less: Control of Volume and Variance in Network Measurement", in Transactions on Information Theory, 2005.

[23] T. M. Chen and S. Liu, "A Model and Evaluation of Distributed Network Management Approaches", IEEE Journal on Selected areas in Communications, vol. 20, no. 4, May 2002.

[24] H. M. Taylor and S. Karlin, *An Introduction to Stochastic Modeling, 3rd Ed.*, Academic Press, MA, USA, 1998.

[25] S. I. Resnick, Adventures in Stochastic Processes, pp. 318-319.

[26] N. Sofra, T. He, P. Zerfos, B. J. Ko, K.-W. Lee, and K. Leung, "Accuracy Analysis of Data Aggregation for Network Monitoring", submitted to the 2nd Annual Conference of ITA (AITAC 2008).