Challenges and Approaches in Providing QoS Monitoring *

Yuming Jiang, Chen-Khong Tham, Chi-Chung Ko
Department of Electrical Engineering
National University of Singapore
10 Kent Ridge Crescent, Singapore 119260
Email:{elejym, eletck, elekocc}@nus.edu.sg

Abstract

The Internet is evolving to provide quality of service (QoS) guarantees to multimedia applications. To ensure that the contracted QoS is sustained, it is not sufficient to just commit resources. QoS monitoring is required to detect and locate the degradation of QoS performance. In addition, the distribution of QoS, instead of simply end-to-end QoS, needs to be monitored. In QoS distribution monitoring, the distribution of QoS experienced by a real-time flow in different network segments is monitored. This paper presents a brief survey of current QoS monitoring related mechanisms, followed by a discussion of the challenges involved in providing QoS distribution monitoring. Several approaches are then proposed to meet these challenges. Finally, the issues that remain open are discussed.

Keywords: Network management, Quality of service, QoS distribution, QoS management, QoS monitoring

1 Introduction

The Internet is evolving to support multimedia applications with diverse performance requirements. To provide quality of service (QoS) guarantees to these applications and ensure that the agreed QoS is sustained, it is not sufficient to just commit resources since QoS degradation is often unavoidable. Any fault or weakening of the performance of a network element may result in the degradation of the contracted QoS. Thus, *QoS monitoring* is required to track the ongoing QoS, compare the monitored QoS against the expected performance, detect possible QoS degradation, and then tune network resources accordingly to sustain the delivered QoS [1, 15]. The mechanisms for QoS monitoring can be classified into two categories, *end-to-end QoS monitoring* and *QoS distribution monitoring* [10], according to the QoS information that can be obtained from them. In an end-to-end QoS monitoring approach, only the end-to-end QoS between the sender and receiver of a real-time flow is monitored.

* This paper was presented in part at 2000 IEEE/IFIP Network Operations and Management Symposium (NOMS 2000), Hawaii, April 2000.

In a QoS distribution monitoring approach, however, the QoS distribution experienced by the flow in different network segments is monitored in addition to the end-to-end QoS.

In recent years, several mechanisms have been proposed for QoS monitoring [4, 16]. In addition, based on the information offered by QoS monitoring, a lot of studies have been conducted to perform further QoS related analysis or operations [6, 7, 13]. However, most of the current QoS monitoring related mechanisms are concerned with end-to-end QoS monitoring or based on the assumption that necessary QoS information, such as QoS distribution information, can be obtained from other mechanisms. Few of them address the problem of QoS distribution monitoring directly. On the other hand, the real-time flow of a multimedia application usually crosses several network segments that provide different levels of QoS. Clearly, to locate the network segment(s) causing possible QoS degradation, QoS distribution monitoring is needed rather than end-to-end QoS monitoring, i.e., the network manager should monitor the distribution of QoS experienced by the flow in different network segments [10].

The purpose of this paper is to identify the challenges involved in providing QoS distribution monitoring. A second goal is to propose possible approaches to meet these challenges. With these approaches, when monitoring a real-time flow, the network manager can locate *relevant monitors* that are metering the flow. By retrieving and consolidating traffic information from these relevant monitors embedded in different network segments, not only the end-to-end QoS but also the QoS distribution of the flow can be derived. In addition, with the monitored QoS distribution, the degradation of QoS can be detected and located. Moreover, the network manager may control the network accordingly to sustain the contracted QoS.

The remainder of the paper is structured as follows. Section 2 presents a brief survey of several QoS monitoring related mechanisms that have been proposed in the literature. The challenges involved in providing QoS distribution monitoring are also identified in this section. Sections 3 and 4 present several approaches to meet these challenges, which include the relevant monitor (RM)-based scheme and improved relevant monitor (IRM)-based scheme for locating relevant monitors in Section 3, and a new method for synchronizing traffic information retrieval in Section 4. Section 5 discusses the issues that remain open, and finally Section 6 offers some concluding remarks.

2 QoS Monitoring

2.1 Model

Figure 1 illustrates the model of a QoS monitoring system based on the traditional network monitoring model presented in [17], which includes the following functional components:

1) *Monitoring application*: This component serves as an interface to the human network manager. Its functions include retrieving traffic information from monitors, analyzing such information and providing analysis results to users. Based on the analysis results, the network manager may perform other operations.

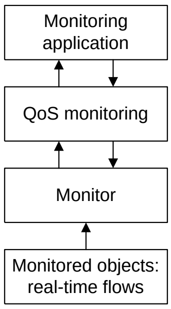

Figure 1: QoS monitoring

2) *QoS monitoring*: This module is an additional one compared with the model of traditional network monitoring systems, which includes all the other three functional components [17]. The new module provides mechanisms for QoS monitoring, which enable the monitoring application to retrieve traffic information from relevant monitors for deriving QoS related parameters.

3) *Monitor*: This module gathers and records traffic information and communicates the information to the monitoring application. In particular, the monitor module performs real-time measurement of real-time flows and reports the measured information to the monitoring application.

4) *Monitored objects*: This is the information such as attributes and activities that are to be monitored in the network. In QoS monitoring, these objects are basically data items (e.g. counts) of monitored real-time flows. A *real-time flow*, corresponding to a multimedia application, is identified by its source and destination addresses at various layers of the OSI (open systems interconnect) reference model [2], which may include its source and destination IP (Internet protocol) addresses, transmission port number such as UDP (user datagram protocol) port number and RTP (real-time transport protocol) SSRC (synchronization source) identifier [16].

The QoS monitoring model is similar to the traditional network monitoring model described in [17] but with the following differences. Firstly, for simplicity, we use one monitor module to represent all functionalities related to traffic information recording. Secondly, the monitored objects are real-time flows in the QoS monitoring model, while in the traditional network monitoring model, the monitored objects are basically the total traffic into and out of the device that a monitor is attached to. Thus, traditional network monitoring is usually limited to the network-layer while QoS monitoring needs to decode application-level protocols (e.g. UDP and RTP). The *application-level* protocols are the higher layer protocols running above the network-layer protocol in the OSI model as defined in [18]. This contributes to the third difference: monitors in the QoS monitoring model must perform application-level monitoring.

The fourth difference is that the QoS monitoring model introduces a new module, the QoS monitoring module, which provides mechanisms for the monitoring application to retrieve traffic information from relevant monitors before deriving QoS related parameters. According to the traffic information retrieved by the mechanisms, we classify them into two categories: *end-to-end QoS monitoring* mechanisms and *QoS distribution monitoring* mechanisms. In an end-to-end QoS monitoring approach, the traffic information is retrieved only from the sender and receiver of a real-time flow. In contrast, in a QoS distribution monitoring approach, the information is retrieved from *relevant monitors* that are metering the flow in different network segments. Clearly, both approaches can detect possible QoS degradation of the flow by consolidating the retrieved traffic information. However, since the flow may cross several network segments that provides different levels of QoS, only the QoS distribution monitoring approach can further isolate the network segment(s) causing the degradation.

Lastly, in addition to the functions supported in the traditional network monitoring model, the monitoring application in the QoS monitoring model may provide further QoS related analysis and operations, such as identifying QoS problems [13], adjusting the monitoring system [6] and reconfiguring the network system [7].

2.2 Related Work

Recently, several mechanisms have been proposed for application-level monitoring and QoS monitoring in the literature.

Waldbusser [18] proposed an extension to the remote network monitoring (RMON) specification, which is referred to as RMON2. With RMON2, an RMON2 probe is not limited to monitoring and decoding network-layer traffic. It can also view higher-layer protocols running on top of the network-layer protocol. In particular, an RMON-2 probe is capable of seeing above the IP layer by reading the encapsulated higher-layer headers such as UDP and RTP. This allows the network manager to perform application-level monitoring required by QoS monitoring.

Brownlee, Mills and Ruth [2] and Brownlee [3] proposed and realized an architecture, referred to as the real-time flow measurement (RTFM) architecture, for the measurement and reporting of network flows. A *flow*, which may be generated by a multimedia application, is defined as "an artificial logical equivalent to a call or connection, belonging to an accountable entity" [2]. In practical terms, a flow is a stream of packets passing through a network between two end points or being sent from a single end point. The accountable entity of a flow is specified by the values of its addresses which may include one or more specific addresses for each layer of the OSI model. In other words, a flow is identified by its source and destination addresses at various layers of the OSI model. Thus, the RTFM mechanism can also provide application-level monitoring.

Chen et al. [4] introduced a software approach to monitoring end-to-end QoS in ATM networks. In order to monitor the QoS of a selected virtual connection, test cells were sent along a parallel connection that had been established with the same route and QoS class as the selected virtual connection. By taking advantage of the fact that ATM switches will statistically multiplex the test cell stream and user cell stream, the test cells would experience a QoS similar to that seen by the selected user connection. Moreover, because the QoS of the test connection could be derived from the information carried by the test cells, the QoS of the selected user connection could be deduced accordingly. A key requirement of this approach is that a parallel test connection with the same route and QoS class be set up to test the selected user connection. Therefore, permanent virtual connections (PVCs) pre-established by the network manager for user connections were used, i.e. the network manager had to know the route of a user connection so that the test connection could be set up accordingly. However, this may be a restrictive requirement since in a real network environment, user connections are usually established dynamically and the corresponding routes may also be dynamic.

Schulzrinne et al. [16] proposed a real-time control protocol (RTCP), which enables RTCP monitors to retrieve traffic information from RTCP messages. Since the RTCP messages of a real-time flow include necessary traffic information about its sender and receiver, the end-to-end QoS parameters of the flow can be derived from these messages directly. Thus, the RTCP monitors can perform end-to-end QoS monitoring. However, since the RTCP messages are generated by the sender and the receiver, only the end-to-end QoS can be monitored.

Mourelatou et al. [13] presented an agent-based approach to identifying QoS problems. The monitoring agents were responsible for monitoring the end-to-end QoS. It was shown that a management system was capable of identifying the cause of performance degradation by correlating the information from these QoS monitoring agents. One key assumption in this approach is that the end-to-end QoS can be monitored and the corresponding information provided by dedicated agents. However, the mechanism through which the agent could monitor the end-to-end QoS was not discussed.

Ehab Al-Shaer [6] proposed an event-driven dynamic monitoring approach for multimedia networks. The task of detecting primitive and composite events is distributed among dedicated monitoring agents as in [13]. An example of an event is the QoS degradation of a multimedia application. Unlike the assumption in [13], this approach requires that prior to any monitoring operation, the system manager must describe the physical or geographical distribution of the multimedia application that the manager intends to monitor. However, it was not shown how the system manager can determine the distribution of a multimedia flow inside the multimedia network.

In summary, various researchers have proposed several QoS monitoring and related approaches from different perspectives and shown that QoS monitoring can be achieved if certain prerequisites or assumptions are satisfied. Waldbusser [18], Brownlee, Mills and Ruth [2] and Brownlee [3] showed that application-level monitoring is possible and feasible. Chen et al. [4] and the RTCP-based scheme showed that end-to-end QoS monitoring can be achieved if the route of a real-time flow is given in Chen et al. [4] or the flow is based on RTP [16]. Mourelatou et al. [13] and Ehab Al-Shaer [6] provide further analysis or operations when QoS degradation has been detected and relevant information has been provided. However, none of these mechanisms addresses the problem of QoS distribution monitoring directly.

2.3 Challenges

Although both end-to-end QoS monitoring and QoS distribution monitoring are capable of detecting possible QoS degradation, it is the distribution of QoS that should be monitored so as to locate possible QoS degradation [10]. Usually, a real-time flow crosses several network segments during its transmission, which may provide different levels of QoS. Clearly, if the QoS seen by the flow receiver is degraded, it is impossible for end-to-end QoS monitoring to locate the degradation. In contrast, the QoS distribution monitoring can provide more information for locating the degradation. However, QoS distribution monitoring imposes the following two challenges.

1) *Locating relevant monitors*: In order to monitor the QoS distribution of a real-time flow, traffic information of the flow needs to be collected from *relevant monitors* that are metering the flow. Since different real-time flows may cross different network segments, the monitors involved in monitoring different flows are different. Thus, a QoS distribution monitoring approach needs to provide mechanisms for the monitoring application to locate relevant monitors of different flows. In addition, since the monitoring application may be mobile, such as Web-based applications, the mechanisms provided should enable the mobile monitoring application to locate relevant monitors and vice versa.

2) *Synchronizing the retrieval of traffic information*: The synchronization of traffic information retrieval among relevant monitors is another challenge in providing QoS distribution monitoring, which includes traffic information recording, reporting and consolidation. To derive QoS related parameters such as loss ratio, traffic information needs to be collected from relevant monitors. This is similar to performance monitoring in distributed systems and also has the so called 'synchronization' problem in retrieving information from distributed measurement points [7, 11]. For example, in the performance measurement system introduced by Lange, Kroeger and Gergeleit [11], the clocks of each measurement points have to be synchronized before the behavior of distributed actions, which start at one node and terminate at another node, could be measured. Similarly, in the QoS monitoring system, traffic information must also be retrieved in a synchronized manner, i.e., the information retrieved from different measurement points must be of the same part of the flow that has been selected for monitoring.

To meet these challenges, a few approaches are proposed in the next two sections.

3 Schemes for Locating Relevant Monitors

In the last section, we argued that to monitor the QoS distribution of a certain flow, traffic information of the flow needs to be retrieved from its relevant monitors. This requires the relevant monitors that are metering the flow to be located. To achieve this, additional mechanisms are needed. Figure 2 shows the model of such a mechanism, which uses a real-time application name server (RTANS) to connect the monitoring application and relevant monitors.

The model illustrated by Figure 2 is based on the QoS monitoring model in Figure 1 with the replacement of the QoS monitoring module by the RTANS. The use of RTANS provides a mechanism for the monitoring application to locate relevant monitors, which operates as follows.

Prior to the monitoring, the real-time application name server (RTANS) is set up and made known to all monitors and the monitoring application in the system.

During monitoring, each monitor registers with the RTANS its address and the traffic attributes of monitored real-time flows. The attributes of a flow include the flow's source and destination addresses, the byte count of the flow's traffic and other attributes. In the RTANS, relevant monitors of each flow

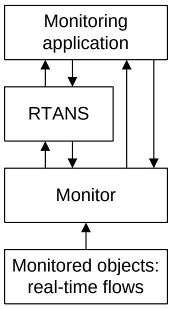

Figure 2: QoS distribution monitoring

are sorted according to the attributes into a real-time application (RTA) table. From the RTA table, the monitoring application can find out the flow's relevant monitors. With the addresses of the relevant monitors, the monitoring application can locate and retrieve traffic information of the flow from them.

Figure 3 and Table 1 illustrate an example of the RTA table. In Figure 3, there are two real-time flows, Flow A from Sender A to Receiver A through Ma, Mb and Mc, and Flow B from Sender B to Receiver B through Md, Mb and Me, where Ma, Mb, Mc, Md and Me are five network devices with embedded monitors. Then, the RTA table generated in the RTANS is as shown in Table 1: Ma, Mb and Mc, and, Md, Mb and Me are involved in monitoring Flow A and Flow B respectively.

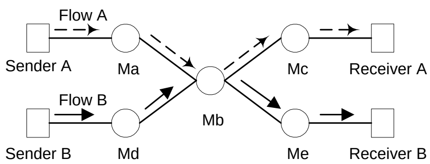

Figure 3: A sample network

Real-time Flow (sender, receiver)	Relevant Monitors
Flow A (Sender A, Receiver A)	Ma, Mb, Mc
Flow B (Sender B, Receiver B)	Md, Mb, Me

Table 1: RTA table

Since this method requires the RTANS to be made known to all the monitors in the system, this mechanism is referred to as the relevant monitor (RM) based scheme [9] and the improved relevant monitor (IRM) based scheme [10]. The following two subsections give overviews of the two schemes respectively. More detailed information about the two schemes can be found in [9] and [10].

3.1 RM-Based Scheme

In the RM-based scheme, there is only one RTANS in the system. All monitors in the system are manually pre-configured to know where the RTANS is. On detecting a new real-time flow, the relevant monitors of the flow register their addresses and the flow's attributes with the RTANS. The RTANS uses the registered information to update its RTA table. During monitoring, the monitoring application searches for the relevant monitors of the flow from the RTA table. After this, it uses the addresses of relevant monitors to locate and communicate with them.

Once the monitoring application locates a relevant monitor, it tells the monitor its address. Then, the monitor will use the address to locate the monitoring application and report traffic information to it.

From the above procedure, the monitoring application can locate and communicate with relevant monitors even if it is mobile, for example, a Web-based application that can be initiated anywhere.

3.2 IRM-Based Scheme

In contrast to the RM-based scheme, there may be more than one RTANSs in the IRM-based scheme, each of which corresponds to a network management domain.

Each monitor in the IRM-based system maintains a RTANS list, which stores the addresses of the RTANSs in the system. When a new domain is set up, the corresponding RTANS is known only within its own domain. The new RTANS needs to register its address with all monitors in the system. Here, the assumption is that the human network manager knows the addresses of all the monitors from which he wants to retrieve traffic information. Based on this assumption, the RTANS is configured by the manager to know the addresses of these monitors and thus it registers its address with all of them. In each monitor, the registration adds the new RTANS to its RTANS list. During monitoring, the monitor uses these addresses to locate and register with all the RTANSs in the list its own address and the monitored traffic attributes of real-time flows.

3.3 Comparison

The functions of the four modules in the QoS distribution monitoring model are similar in both the RM-based and IRM-based schemes, but there are three principal differences:

1) In the RM-based scheme, there is only one common RTANS for the whole network, while in the IRM-based scheme there are more than one RTANSs, each for a different network management domain.

2) In the RM-based scheme, each monitor is manually configured to know where the only RTANS is. In contrast, in the IRM-based scheme, each RTANS registers its address with all monitors. The manual configuration is made in the RTANS.

3) In addition, in the IRM-based scheme, each monitor can report traffic information to multiple monitoring applications that belong to different network management domains.

These differences contribute to the improvement of the IRM-based scheme over the RM-based scheme. With the IRM-based scheme, not only can relevant monitors be located and traffic information retrieved, but these operations can be performed by monitoring applications from different network management domains. When there is only one network management domain, the RM-based scheme is preferred because it needs less programming in implementing the monitor module. However, if more than one network management domains exist in the network, as is usually the case, it is the IRM-based scheme that should be adopted.

4 An Approach to Synchronizing the Retrieval of Traffic Information

The synchronization of traffic information retrieval is another challenge in providing QoS distribution monitoring. In order to derive QoS related parameters of a real-time flow in different network segments and compare them to detect and locate possible QoS degradation, traffic information of the flow needs to be retrieved from its relevant monitors residing in these network segments. In addition, the retrieval of the information must be performed in a synchronized manner. That is, the information retrieved and used to derive the QoS parameters must be that of the same part of the flow. By the same part of the flow, we mean the traffic of the flow within the same time interval since the flow is seen by each monitor at its measurement point.

A common approach to the synchronization problem in distributed measurement systems is to synchronize all monitors to a certain reference time such as the coordinated universal time (UTC). This can be achieved by using either the Global Positioning System (GPS) [5], a global time base (GTB) [11], or the Internet network time protocol (NTP) [12]. However, additional hardware is required to receive time signal from the GPS [5] or to implement the GTB [11]. Furthermore, synchronization may not be accurate for QoS monitoring in a large scale network if NTP is used, due to various network delays [12].

This paper adopts another method to synchronize the retrieval of traffic information. It is based on the *start times* of a real-time flow seen by its relevant monitors. The *start time* is the time when the *start packet* of the flow reaches a measurement point. The *start packet* is the first packet of the flow sent by its sender. Clearly, the start time seen at one measurement point is different from another measurement point due to various network delays. Also due to the uncertainty of network delays, the traffic information reported by different monitors may reach the monitoring application at different times. In the worst case, the information from some monitors may be lost during the transmission. Thus, mechanisms for recording, reporting and consolidating traffic information are needed to ensure that traffic information retrieval is synchronized among relevant monitors.

4.1 Recording Traffic Information

In [2], Brownlee, Mills and Ruth proposed a real-time flow measurement (RTFM) architecture for the measurement and reporting of network traffic flows. This can be adopted by the monitor module in the

QoS monitoring model shown in Figure 2 with some slight changes.

Under this architecture, the recorded attributes of each flow include: (1) the addresses for the flow's source and destination; (2) the start timestamp when the first packet was seen by the monitor; (3) the byte count of the flow's traffic, etc.

The address attribute is used to identify different flows. In the case that two recorded flows have the same address attributes, e.g. one old flow has stopped and one new flow is being sent, the start timestamp attributes are used to identify them. The monitor uses a rolling counter to count the flow's traffic, to which the byte count of subsequent traffic on the same flow is added. The byte count stores the number of bytes of the flow that has been counted since it was detected by the monitor. In the RTFM architecture, the start timestamp records the time when the monitor detected the first packet of the flow. To use the RTFM monitor in the QoS monitoring model, we propose a change.

We propose that the start timestamp store the *start time* when the *start packet* of the flow reached the measurement point, which is the first packet sent by the flow's sender. If the start packet is not lost before arriving at the measurement point, the start time is exactly the time when the monitor detected the first packet of the flow. If the start packet has been lost before the measurement point, the monitor needs to estimate the arrival time of the start packet, the start time, as if it had not been lost. In order to do this, the arrival interval between the lost start packet and the first detected packet needs to be estimated. We use the arrival interval between the first and second detected packets to estimate it on the assumption that the network status is unlikely to change during the two intervals. Denote by t_0^i the start time of flow i, the timestamps of the first and second detected packets by t_1^i and t_2^i respectively, and the *sequence number*s of the start packet, the first and the second detected packets by s_0^i, s_1^i and s_2^i respectively. We have: $t_0^i = t_1^i - (s_1^i - s_0^i) * \frac{t_2^i - t_1^i}{s_2^i - s_1^i}$. Here, we assume that the protocol such as RTP [16] used by the flow supports sequence numbers, where each packet is labeled with a *sequence number* that increases by one. In addition, we also assume that the *initial sequence number* s_0^i of the start packet is known to the monitor.

In the case that the initial sequence number is random as recommended by RTP [16], it is impossible to determine whether the first detected packet is the start packet and how many packets are between the first detected packet and the start packet. Thus, the start time of the flow cannot be derived. In order to synchronize the reporting of traffic information and overcome the uncertainty of the initial sequence number of the start packet, we use the following method to determine the start time of the flow. Consider that real-time flow i is measured by several monitors $M_k, k \in (a, b, c, \ldots)$. The *least first sequence number* of the first detected packets of the flow by all monitors is $s_{min}^i (= min[s_1^i(M_k), k \in (a, b, c, \ldots)])$, where $s_1^i(M_k)$ denotes the sequence number of the first detected packet by monitor M_k. Then, the least first sequence number is considered to be the initial sequence number of the flow, and thus the start time of flow i in monitor M_k, denoted by $t_0^i(M_k)$, is determined by: $t_0^i(M_k) = t_1^i(M_k) - (s_1^i(M_k) - s_{min}^i) * \frac{t_2^i(M_k) - t_1^i(M_k)}{s_2^i(M_k) - s_1^i(M_k)}$. The least first sequence number is determined in the RTANS by comparing the sequence numbers of the first detected packets by all relevant monitors of the flow.

Note that in the above description, we assume that the protocol used by a real-time application supports sequence numbers and this protocol can be detected by RTFM meters. However, this is not usually the case. For example, when the flow is an aggregate of differentiated services flows, it does not have sequence numbers in its packets. In these cases, the start timestamp is just the time when the first packet was detected by the monitor as proposed in the RTFM architecture. In addition, the *first sequence number* of the flow is set to zero.

4.2 Reporting Traffic Information

The *start times* observed at different measurement points are used to synchronize the reporting of traffic information of a real-time flow by its relevant monitors.

Each relevant monitor reports to the monitoring application the traffic information of the flow in the same reporting interval, which started from the start time of the flow seen by this monitor. Denote by $\tau^i(M_k)$ the time a relevant monitor M_k receives the retrieving request from the monitoring application. Denote by Δ the reporting interval. Clearly, there is an integer L, referred to as the *interval sequence number*, such that $(t_0^i + (L-1) * \Delta) < \tau^i(M_k) < (t_0^i + L * \Delta)$. The monitor starts reporting at time $t_0^i + L * \Delta$ and repeats it in the reporting interval Δ. Thus, the traffic information reporting is synchronized in the time scale among relevant monitors (see Figure 4). Compared with the use of the time when the flow was detected by the monitor, using the start time as the synchronization basis is more reliable under the situation that the first packets seen by different monitors may be different due to network loss.

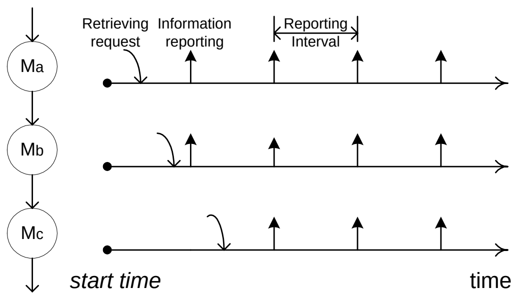

Figure 4: Synchronization of reporting

The information of each flow registered to the RTANS includes its *address attribute*, its *start time*, the *first sequence number* of the first detected packet and the *monitor's address*. The RTANS sorts the relevant monitors of each flow, which is identified by its address attribute and start time, to generate the real-time application (RTA) table. Also, the RTANS determines the *least first sequence number* of the flow by comparing the *sequence numbers* of all the *first* detected packets of the flow. The RTA table helps the monitoring application to find the addresses of the relevant monitors of the flow that is selected to monitor and to locate these monitors to retrieve traffic information of the flow. Before traffic information retrieval, the monitoring application should obtain the *least first sequence number* information from the RTANS and send it to these monitors, which will be used in each monitor to determine the *start time* for traffic information reporting.

During the traffic information reporting, each relevant monitor reports traffic information of the flow to the monitoring application periodically in the reporting interval. The reported information consists of the *byte count* by the end of each reporting interval and the *interval sequence number* starting from one and increasing by one. The interval sequence number will be used in consolidating traffic information in the monitoring application, which is described in the next subsection.

4.3 Consolidating Traffic Information

In the last subsection, we have considered the synchronization problem of traffic information reporting. However, even if the traffic information reporting is synchronized, the times at which the information from different monitors arrives at the monitoring application may still be different because of network delays. Two principal delays contribute to the difference. One is the transit delay for the flow to transfer from one measurement point to another measurement point. The other is the transmission delay of the traffic information from each monitor to the monitoring application.

In order to consolidate traffic information from different monitors, the monitoring application uses a buffer to store reported traffic information of the flow from all relevant monitors. It keeps monitoring the buffer until the information with the same *interval sequence number* from all relevant monitors has been received or the information with the next interval sequence number starts arriving. After this, the monitoring application uses the retrieved information to derive QoS related parameters. At the same time, it removes the traffic information of this reporting interval from the buffer and waits for the next reporting interval information.

Clearly, through the above-mentioned procedures for recording, reporting and consolidating traffic information, the retrieval of traffic information from relevant monitors is synchronized and thus QoS related parameters can be further derived in the monitoring application based on the retrieved information.

5 Discussion

So far, we have introduced two schemes for locating relevant monitors and a new approach for synchronizing traffic information retrieval. The two schemes have been implemented as prototypes using CORBA [14] technique, which are described in [9] and [10] respectively. In addition, the synchronization approach has also been adopted in designing QoS monitoring systems [8]. However, several issues remain open at this time.

1) *Selection of reporting interval:* An important issue which remains open concerns the length of a "good" reporting interval. Although a smaller interval gives more detailed information, it comes at a certain cost: the smaller the reporting interval, the larger the management traffic caused by traffic

information reporting.

2) *Estimation of lost information:* Packet loss is a natural event in communication networks and it affects QoS monitoring. A QoS monitoring system has to deal with at least two kinds of packet loss. One is the loss of the *start packet* of the flow that has been selected for monitoring. The second is the loss of traffic information reported by its relevant monitors. Thus, the problem of how to estimate the lost information arises. To deal with the lost start packet, the times of the first two detected packets have been used to estimate the *start time* at which the start packet was supposed to reach the measurement point. To estimate lost traffic information for deriving QoS parameters in its corresponding reporting interval, one possibility is to use the last interval traffic information from the same monitor. Clearly, the problem of how good these estimations are arises and further study is needed.

3) *Scalability:* Scalability is another important issue for a QoS monitoring system. We believe that many other mechanisms can be integrated with the mechanisms described earlier to improve the system scalability. For example, hierarchical monitoring and flow aggregate monitoring are two of them which will be discussed below. However, further experiments are needed to verify this.

4) *Hierarchical monitoring:* Hierarchical monitoring is an important approach to improve a monitoring system's scalability such as in RMON2 [18] and in Ehab Al-Shaer [6]. We believe that it can also be integrated to QoS distribution monitoring systems. For example, relevant monitors are arranged hierarchically in the RTANS. The monitoring application first retrieves traffic information only from the relevant monitors on the top layer. After detecting and locating the degradation of QoS performance, the monitoring application can further communicate with the monitors below the two monitors in the monitor hierarchy which detected the degradation.

5) *Monitoring flow aggregates:* In a wide area network, there may be a large number of real-time flows. Thus, a QoS monitoring system should be capable of handling all of these. One way is to monitor flow aggregates. A flow aggregate is a combination of individual flows with similar attributes. An example of a flow aggregate is an aggregate of differentiated services flows. In this paper, a flow is defined as a logical connection as in [2] and is identified by its source and destination addresses which may include one or more specific addresses for each OSI layer. Clearly, if a flow aggregate is the combination of individual flows that have the same source and destination addresses at a certain OSI layer, it can be considered as a "pipe" flow. In addition, since the mechanisms introduced in this paper so far are per-flow based, they should also be applicable for monitoring flow aggregates. However, since the pipe flow is combined by several individual flows, it may not have explicit sequence numbers. Therefore, it may be impossible to determine its start time and thus the arrival time of the first detected packet should be used as the synchronization basis rather than the supposed arrival time of the start packet.

6) *Monitoring delay and delay variation:* Throughput, loss, delay and delay variation are four important QoS parameters. However, since the recorded and retrieved traffic information are mainly the start time and byte counts, only the throughput and loss can be derived [8]. To derive delay and delay variation, the recorded and retrieved traffic information must include other timing information.

In particular, the time of each packet arrival at a measurement point needs to be recorded and reported. In addition, for the derivation of delays, all monitors must be synchronized as in [11]. However, timestamping each packet and synchronizing clocks of all monitors will increase the system complexity. In addition, timestamping every packet may cause monitors to exhaust their processing ability quickly and hence reduce the system scalability. The third problem for monitoring delay and delay variation concerns the selection of packets. Because of the restriction of management traffic, it is not feasible to monitor delay or delay variation for each packet. Thus, packets should be selected statistically to estimate them. Hence, for monitoring delay and delay variation, further investigation is needed. Nevertheless, we may expect that in the future Internet, packets from real-time applications will be dropped in the network if they cannot meet their delay requirements. Hence, from the packet loss information, such delay vitiation could be located.

6 Summary

Providing sustained QoS to multimedia applications imposes requirements on QoS monitoring. In this paper we have identified issues involved in providing QoS monitoring and briefly surveyed research in this area. We identified two challenges towards providing QoS distribution monitoring, which are the difficulties in locating relevant monitors and in synchronizing traffic information retrieval. Several approaches have been proposed to meet these two challenges. However, many issues remain open at this time, which require further investigation. Nevertheless, the approaches proposed in this paper can be adopted directly in monitoring many QoS parameters such as throughput, loss rate and loss ratio [8]. In addition, they should be helpful in monitoring delay and delay variation as well. For example, we may expect that future network devices will be able to monitor and record their own behavior, such as delay of each flow passing through them, as proposed by Chen et al. [5]. Under this condition, the RM-based and IRM-based schemes may be used to retrieve delay related information from the monitors embedded in these network devices.

References

[1] C. Aurrecoechea, A. Campbell, and L. Hauw, "A Survey of QoS Architectures", *Multimedia Systems Journal* , vol. 6, no. 3, pp. 138-151, May 1998.

[2] N. Brownlee, C. Mills, and G. Ruth, "Traffic Flow Measurement: Architecture", IETF RFC 2063, 1997.

[3] N. Brownlee, "Traffic Flow Measurement: Experiences with NeTraMet", IETF RFC 2123, 1997.

[4] T. M. Chen, S. S. Liu, M. J. Procanik, D. C. Wang, and D. D. Casey, "INQIRE: A Software Approach to Monitoring QoS in ATM Networks", *IEEE Network*, pp. 32-37, March/April 1998.

[5] T. M. Chen et al., "Monitoring and Control of ATM Networks Using Special Cells", *IEEE Network*, pp. 28-38, Sept./Oct. 1996.

[6] Ehab Al-Shaer, "Hierarchical Filtering-based Monitoring Architecture for Large-Scale Distributed Systems", PhD Dissertation, Computer Science Department, Old Dominion University, July 1998.

[7] L.J.N. Franken and B.R. Haverkort, "The Performability Manager", *IEEE Network*, pp. 24-32, Jan./Feb. 1994.

[8] Y. Jiang, "Monitoring QoS Distribution in Multimedia Networks", Technical Report, Department of Electrical Engineering, National University of Singapore, 1999.

[9] Y. Jiang, C. K. Tham, and C. C. Ko, "A Web-based Real-Time Traffic Monitoring Scheme Using CORBA," *2nd IFIP/IEEE Int. Conf. on Management of Multimedia Networks and Services*, France, Nov. 1998.

[10] Y. Jiang, C. K. Tham, and C. C. Ko, "A QoS Distribution Monitoring Scheme for Performance Management of Multimedia Networks", *Proc. of IEEE GLOBECOM'99*, Brazil, Dec. 1999.

[11] F. Lange, R. Kroeger, and M. Gergeleit, "JEWEL: Design and Implementation of a Distributed Measurement System", *IEEE Trans. Parallel and Distributed Systems*, vol. 3, no. 6, pp. 657-671, Nov. 1992.

[12] D. Mills, "Internet Time Synchronization: The Network Time Protocol", *IEEE Trans. Commun.*, vol. 39, no. 10, pp.1482-93, Oct. 1991.

[13] K. E. Mourelatou, A. T. Bouloutas, and M. E. Anagnostou, "An Approach to Identifying QoS Problems", *Computer Communications* 17, pp. 563-570, 1994.

[14] OMG, *The Common Object Request Broker: Architecture and Specification* v2.0, 1996.

[15] G. Pacifici and R. Stadler, "An Architecture for Performance Management of Multimedia Networks", *IFIP/IEEE Int. Symp. on Integrated Network Management*, May 1995.

[16] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a Transport Protocol for Real-Time Applications", RFC1889, 1996.

[17] W. Stallings, *SNMP, SNMPv2 and RMON : Practical Network Management*, 2nd edition, Addison-Wesley, 1996.

[18] S. Waldbusser, "Remote Network Monitoring Management Information Base Version 2 using SMIv2", RFC2021, 1997.