マルチメディア通信と分散処理 70－7
（1995．5．25）

分散型マルチメディアシステムSymphonyにおける品質保証機構

呉 世雄* 島村 栄* 椿野 慎治†
藤川 和利† 下條 真司‡ 松浦敏雄* 宮原 秀夫*

* 大阪大学基礎工学部情報工学科 † 奈良先端科学技術大学院大学情報科学研究科
‡ 大阪大学大型計算機センター

Email: osw@ics.es.osaka-u.ac.jp

本論文では，分散型マルチメディアシステムSymphonyにおけるQoS(Quality of Servide)アーキテクチャの設計と実装について述べる。このアーキテクチャは，ユーザー、アプリケーション、ネットワークの三層からなり、異なる層の間でアプリケーションQoSマネジャ、ネットワークQoSマネジャとトータルQoSマネジャが、層間のQoSの変換および下位層からのモニタリングと通知を行なう。また，ネットワーク層はbest effort, compulsory, threshold型の品質保証に対応して上位層に対応したQoS制御を行なう。最後にbest effor型でのSympohonyの実装について述べる。

The QoS architecture in the distributed multimedia system Symphony

Seiwoong Oh* Hisashi Shimamura* Shinji Tsubakino†
Kazutoshi Fujikawa† Shinji Shimojo‡ Toshio Matsuura*
Hideo Miyahara*

* Department of Information and Computer Science
Faculty of Engineering Science, Osaka Univ.
† Graduate School of Information Science,
Nara Institute of Science and Technology
‡ Computation Center, Osaka Univ.

Email: osw@ics.es.osaka-u.ac.jp

Described in this paper is the design and implementation of the QoS architecture in the distributed multimedia system Symphony. Our architecture contains the application QoS manager, the network QoS manager between different layers. And also, we design total QoS manager to find a bottleneck resource. Our architecture can support best effort, compulsory and threshold as QoS guarantee types. And also we show the current implemented QoS architecture of the Symphony.

1 Introduction

Recent technological developments in high speed networks and multimedia workstations are making possible distributed multimedia applications such as desktop video conference, distance learning, remote multimedia database access. Mosaic(a client system of WWW(World Wide Web)[1]), nv and vat(a transmission system for video and audio respectively) are examples of popular multimedia application in wide area network.

However, in current multimedia application, multimedia information is played back 'as is' basis on various system environment such as network, server/client machines, and operating system. Therefore, an important information on video material may be skipped or an important portion of a lecture can not be heard. In worst case, author's original intention may be lost.

The above problem is basically due to the overhead of computer and network, 1and to unpredictable responsiveness. Furthermore existing OS(UNIX) do not support latency, throughput, real time responsiveness. And also existing communication architecture(TCP/IP) is based on a best effort performance model and was not designed to support QoS(Quality of Service). In the internet the support of reliable data transfer was a primary goal and performance QoS was only a marginal consideration.

The recent studies on some methods to manage the given system resource have been performed for solving these problems. Real time OS has been studied for the responsiveness guarantee at OS level[2][3]. And also some studies contains the intended QoS degradation when a multimedia application needs more capacity of the cpu speed[4][5]. Some communication protocols for supporting various QoS have been studied as network level[6].

Therefore several QoS architectures[7], which is used for definition and integration of services and mechanism for performance QoS at each levels, have been studied for wide range of multimedia applications as its primary goal.

In this paper, we describe the design and implementation of the QoS architecture of the Symphony supporting various QoS guarantee types. And also we show the current implementation of the QoS architecture in the Symphony.

The remainder of this paper is organized as follows. Section 2 contains the QoS architecture in the Symphony. In Section 3, the implementation of the QoS architecture in the Symphony is described and finally Section 4 briefly concludes the paper.

2 The QoS architecture in the Symphony

In this section, we show the layered structure as the QoS architecture in the Symphony, and the functionality of QoS managers. And also the supported QoS guarantee types are described.

2.1 A layered QoS structure

Fig. 1 illustrates the QoS architecture of services and mechanisms for QoS management and control of multimedia data in the Symphony.

In functional term, the QoS architecture illustrated in Fig. 1 is broadly divided into

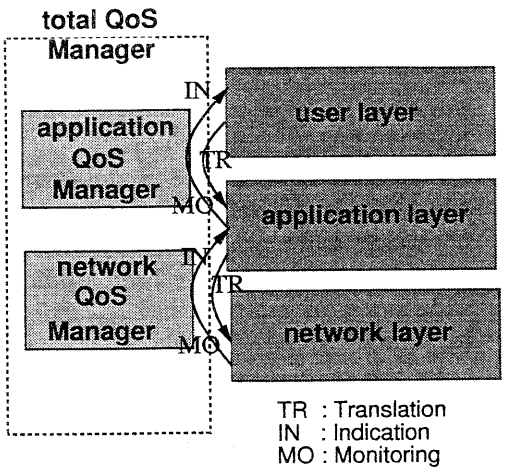

Figure 1: the QoS architecture in the Symphony

a number of layers.

In the uppermost layer, called the user layer, it receives user requirements through scenarios provided by authors(information provider), and pass them from user to the application layer.

The middle layer, called the application layer, performs multimedia applications augmented with services to provide the orchestration while in the lowest layer, called the network layer, network communication is provided.

The application QoS manager, as shown in Fig. 1, translates QoS requirements from the user layer into a set of QoS parameters that can adequately describe QoS requirements of the application layer. And also it monitors the condition of the application layer and can indicate its inability to maintain the required QoS at the application layer to the user layer for re-negotiation.

The network QoS manager translates QoS requirements from the application layer into QoS parameters accepted by the network layer. Besides it monitors the network condition and indicates the network QoS degradation to the application layer for re-negotiation.

The total QoS manager finds a bottleneck resource from QoS information records of each QoS managers and reports the results to each QoS managers if it is required.

2.2 The Application QoS Manager

In the application layer, QoS degradation may be inevitable when system resources are not sufficient. QoS degradation basically decrease the amount of multimedia data processing. When the degradation is performed, the priority among media and QoS parameters of media must be considered at first. In the Symphony, the policy[4] determines it depending on the purpose of the multimedia application. For example, the priority of audio may be higher than the others in TV news while that of video may be higher in Tennis lesson.

The policy designates the synchronization types as well as among media and QoS parameters of media for QoS degradation. As shown in Fig. 2, synchronization types are composed of "skip", "delay", "stop". In "skip" type, each medium in a scene is displayed in real time fashion. Each medium is displayed without losing any contents by changing the playing back rate in "delay" type. Finally, in "stop" type, each medium is displayed without losing any contents admitting a pause.

Media scaling is a kind of QoS degradation manipulating some QoS parameters

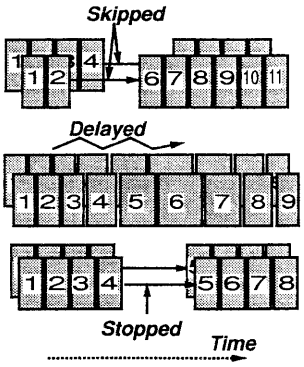

Figure 2: Synchronization types

Figure 3: The media scaling varying the coeeficient of compress

including inter-media and intra-media[4]. Latency, temporal crops. spatial crops, continuity, synclag and so on are considered in the Symphony. Temporal crops are composed of sample rate (audio) and frame rate(video) while spatial crops are the number of bits per sampling(audio) and the tone of color, size, the coeeficient of compress. Fig. 3 illustrates an example of media scaling varying the coeeficient of compress.

2.3 The Network QoS Manager

The requirements of the application layer are translated into a set of QoS parameters composed of maximum message size, rate or throughput and maximum delay and delay jitter[6] through the network QoS manager.

The network QoS manager supports QoS re-negotiation for which the monitoring and indication mechanism are required. The purpose of monitoring tests and verifies QoS during the requested QoS is serviced. If the QoS is not satisfied, the network QoS manager indicates it to the application layer.

For the above, the network QoS manager has the ability to start and stop the flow of media data on sets of related connections, and monitors the on-going temporal relationship between related connections.

2.4 Supporting various QoS guarantee types

In the Symphony, The considered QoS guarantee type is as follows.

best effort the network will do its best to reach the QoS required by the application layer. Nothing will be done in case the network is unable to maintain the required QoS parameter.

compulsory if a compulsory QoS is not satisfied after its negotiation, then the network must reject the connection.

threshold after its negotiation, the network QoS manager will have to monitor the network and to issue a special indication primitive to the application QoS manager whenever it notices its inability to maintain the requested threshold values.

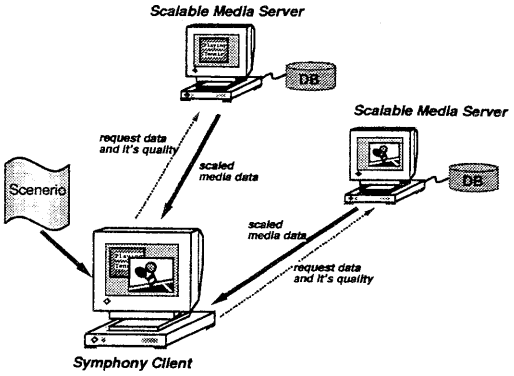

Figure 4: The overview of the Symphony

3 The Implementation of the QoS architecture in the Symphony

A distributed multimedia information service consists of multimedia servers and a client connected with network as shown in Fig. 4. The Symphony is running on a client under the environment like Fig. 4.

3.1 The Application QoS Manager

The application QoS manager is performed in SQUASH (Symphony Quality Adaptation Strategy Handler) as shown in Fig. 5. As described in 2.2, it performs QoS degradation depending on the policy. It receives the QoS requirement from the user layer through a scenario. It monitors Media Process for checking if the required QoS is satisfied or not at the application layer.

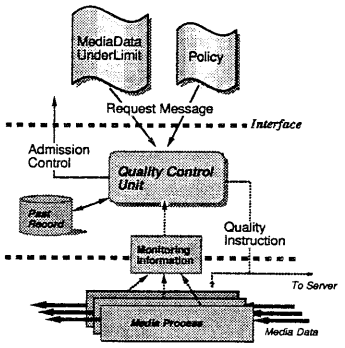

Figure 5: SQUASH in the Symphony

3.2 The Network QoS Manager

UDP/IP is adopted as network protocol of the current Symphony. Since UDP/IP supports best effort model, it has not been implemented yet.

3.3 Total QoS Manager

Since the current Symphony can support only best effort type, the network QoS manager is not implemented yet. Only the application QoS manager records QoS informations. Therefore total QoS manager of the current Symphony is implemented in SQUASH for finding a bottleneck resource of the application layer.

SQUASH do not consider which resources is a bottleneck but only records the status of each buffers in Media Process. Total QoS manager determines which resource is a bottleneck from QoS informations of SQUASH. After it determines a bottleneck resource, it reports the result to SQUASH for adaptive QoS controls.

As shown in Fig. 5, for example, the transmitted media data from server is

buffered in Media Process. If the buffer is overflowed, it represents that the client's actual processing rate is not sufficient. Otherwise if the buffer is underflowed, it represents that the processing power of client is enough. SQUASH considers the priority of the media using the reported information about the bottleneck resource before QoS degradation is performed.

4 Conclusion

The QoS architecture of the Symphony is presented. Our architecture contains the application QoS manager, the network QoS manager between different layers. And also, we design total QoS manager to find a bottleneck resource.

The application QoS manager translates QoS requirements from the user layer and monitors the condition of the application layer. When the application layer can't maintain the requested QoS, the application QoS manager indicates it to the user layer for re-negotiation.

The Network QoS manager translates QoS requirements from the application layer and monitors the network condition and indicates the deterioration of network to the application layer for re-negotiation. Besides it can support best effort, compulsory and threshold as QoS guarantee types.

And also we showed the current implemented QoS architecture of the Symphony.

Our future work contains the implementation of the network QoS manager. For the purpose of that, adoption of real time protocols in stead of the UDP/IP is required.

References

[1] Berners-Lee,T., et.al., *"World-Wide Web: The Information Universe"*, Electronic Networking: Research, Applications, and Policy, Vol.1 No.2, Spring, 1992.

[2] Tokuda,H., et.al., *"Real-Time Mach: Towards a Predictable Real-Time System"*,Proceedings of the USENIX 1st Mach Symposium, Oct.1990.

[3] Robin,P., et.al., *"Implementing a QoS Controlled ATM based Communications System in Chorus"*, Internal Report No. MPG-94-05 Department of Computing, Lancaster University, March 1994.

[4] Fujikawa,K.,Shimojo,S.,Simamura,H., Teranishi,Y.,Matuura,T., and Miyahara,H., *"Application Level QoS Modeling for A Distributed Multimedia System"*, Proceedings of the 1995 Pacific Workshop on Distributed Multimedia Systems, pp.44-51, Hawaii, March, 1995.

[5] Delgrossi,L., et.al., *"Media Scaling for Audiovisual Communication with the Heidelberg Transport System"*, ACM Multimedia '93, pp.99-112, August, 1993.

[6] Topolcic,C., *"Experimental Internet Stream Protocol, Version 2 (ST-II)"*, RFC1190, October, 1990.

[7] Campbell,A., et.al., *"A Quality of Service Architecture"*, Computer Communication Review, Vol. 24, No. 2, Apr. 1994.