Quality of Service issues in multimedia systems

By
Sandesh P. Meda
Department of Computer Science, University of Texas at Arlington

---

1. Overview

In multimedia systems, the notion of Quality of Service (QoS) assumes importance. Multimedia applications have requirement in terms of bandwidth, delay and delay jitter and to specify these requirements, the application uses the notion of QoS. The need to define QoS arises from the realization that the users require different quality presentations at different times. [4]

QoS notion, mechanisms to specify QoS requirements, QoS negotiation and renegotiation, admission control and resource management is important in multimedia system. If the application is real time these issues have to be given top priority and are critical. This paper describes an overall framework for a Multimedia System and discusses the various issues and means and mechanism to manage them. Mechanisms and means are identified to deal with the issues to provide guaranteed QoS to the End User. QoS must be specified and guaranteed end-to-end at all levels. [4]

2. Introduction

Multimedia systems are real-time systems: they must perform tasks and deliver results according to a schedule that is externally determined. The degree to which this is achieved by the underlying system is known as the QoS enjoyed by an application. There is no universally accepted definition for QoS. Intuitively, however, QoS defines characteristics that influence the perceived quality of an application. However, the following definition of QoS holds good.

"QoS is a quantitative and qualitative specification of an application's requirement which a multimedia system should satisfy in order to achieve desired application quality" [4]

Based on this definition, there are two aspects to QoS. Firstly, applications specify QoS requirements and the systems provide a QoS guarantee. To be able to specify QoS aspects concisely, it must be specified as a set of parameters that can be assigned numerical values. In a multimedia presentation, the ultimate user of the system is a human being. Thus, the quality of the presentation is a matter of the user's perception. Thus, it is necessary to specify a range of values rather than a single value.

As the time progressed, Multimedia systems have become more and more complex in terms of the services they provide and the management of the underlying physical infrastructure that implements these services. The situation gets further complicated if the user requirements are taken under consideration in addition to already present inherent complexity of the Multimedia systems. [6]

3. Overall Quality of Service (QoS) Framework

A simplified QoS operation model of a multimedia communication system can be described as follows. The user's requirement is specified via an Application Programming Interface(API) [6] The system then determines whether it has sufficient resources to meet the requirements. If so, it will accept the application and allocate the necessary resources to serve the application so that its requirements are satisfied. If it has insufficient resources to meet the application's requirement, it may either reject the application or suggest a lower QoS requirement that it can satisfy to the application. On the basis of this operation model, there should be the following components in order to provide QoS guarantees: [4]

- A QoS specification mechanism for application to specify their requirements.
- Admission control to determine whether the new application should be admitted without affecting the QoS of the other on going application.
- A QoS negotiation process so that a system may support as many application as possible.
- Resource allocation and scheduling to meet QoS requirement of accepted applications.
- Traffic policing to make sure that applications generate the correct amount of data within agreed specification.

The model considered here is fairly simple which tends to get complicated when the QoS requirements change during an application session. Sometimes the negotiated parameters cannot be maintained due to network congestion, requiring renegotiation. Some parameters are mutually dependent or contradictory, an example of this is decreasing the error rate by permitting the retransmission will increase the average transmission delay. In spite of the contract resulting from QoS negotiation, the actual QoS values in a System can vary over the time. Therefore a system must continuously monitor the actual QoS and employ correction mechanism like blocking low priority tasks. In this perspective maintaining QoS becomes a complex problem.

4. QoS Specification

To provide QoS specification and guarantees, a connection-oriented session should be used. Before the session is established, QoS parameters are specified and negotiated among all subsystems concerned.

In considering the QoS model described in the previous section, it is convenient to consider three conceptual layers, as shown in the figure 1. [6]

4.1 User Layer

The highest layer is the User Layer. At this level, QoS is specified qualitatively. The user may specify types of application and desired quality. The perceived quality is to some extent subjective. The user's chosen quality is also related to service charges : the higher the required quality, the higher the charge. This policy will discourage users from always using the highest quality. The users are the starting point for an overall consideration of QoS. Thus the primary source of QoS requirements is the user and a suitable interface (an application programmer interface or an API) should be provided to facilitate the choice of parameters. At this level, the detailed system QoS parameters are often meaningless to the user and should be hidden. A better approach is to present choices from examples of varying quality, such as video of normal TV or HDTV quality, or speech of telephone or CD quality. The user's choices are then automatically

mapped onto application parameters. The interface should also store user profiles to avoid making the user repeat the potentially lengthy selection process. [6]

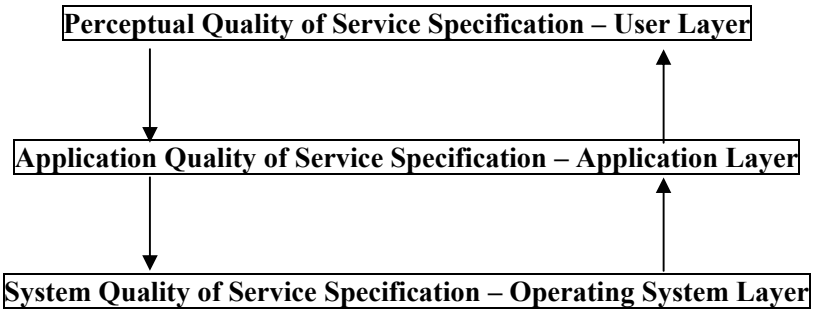


Figure 1 : Conceptual Model of QoS

4.1.1 QoS Parameters

The QoS parameters in a Multimedia System can be categorized into the following five categories[8]

Category	Example Parameters
Performance Oriented	Delay Bit rate
Format Oriented	Video Resolution Frame Rate Storage Format Compression Scheme
Synchronization Oriented	Skew between the beginning of Audio and Video sequences
Cost Oriented	Connection and data transmission charges Copyright fees
User Oriented	Subjective image Sound Quality

Some of the parameters mentioned above are discussed below[8.]

Delay : This parameter specifies the total delay experienced by frames from traveling from the sender to the receiver. In some real time applications, data with acceptable level of error is acceptable than delayed data.

Bit Rate : This parameter would be used for resource reservation. This is measured in number of kilobits per second

Data quality: The data transmitted in a Multimedia System can be Video or Acoustics. Subjective image and sound quality needs to be taken care of. The sampling rate in case acoustic data and the resolution in case of video data needs to be given due importance depending on the kind of application.

Commitment: Best effort or guaranteed can be specified. There are two types guarantees that can be provided: statistical, or deterministic.

Frame Losses: The number of frames lost during the transmission of frames from sender to the receiver. Frame losses while transmission is acceptable to a limit set for a particular application. The user sets this criterion. In some sensitive application like in the field of medicine, these losses of frames during cannot be tolerated.

4.1.2. End-to-End QoS levels

The specification of QoS parameter values that were described above determines the types of service. There are at least three types of services distinguished. [6]

4.1.2.1 Guaranteed Services provide QoS guarantees as specified through the QoS parameter values. The deterministic QoS parameters can be represented by a real number at certain time, it means:

$$QoS : T \rightarrow R$$

Where T is a time domain representing the lifetime of a service, during which QoS should hold and R is the domain of real numbers representing the value of the QoS parameter. The overall QoS deterministic bounds can be specified either by a single value or a pair of values [QoS_{min}, QoS_{max}]

The guaranteed service is also known as Hard QoS [6]

4.1.2.2 Best Effort Service says that the system will try its best to support the application but it does not offer any guarantees. In other words it offers the basic connectivity with no guarantee of delivering of information from node to node. It is also sometimes referred to "Lack of QoS". [6]

4.1.2.3 Predictable Service is based on past network behavior, hence the QoS parameter bounds are estimates of past behavior which the service tries to match. For example: if bandwidth $B_n^{predict}$ was calculated as an average of the bandwidth which the service provided in the past, then the predictable service could promise to provide bandwidth B_n with $B_n <= B_n^{predict}$

The type of service selected depends upon the application of the user. Each of the service is suitable for a particular application though the user can migrate from one service to the other. The type of service selected also depends on cost of employing the services. The cost of the Guaranteed Service is more than that of best effort service. [6]

4.2 Application Layer

At this level, a user's chosen application and quality requirements are mapped (translated) onto a set of parameters that the application level must satisfy in order to meet the user's requirements. The parameters at this level are concerned with media logical data units, such as video frames and audio samples. For example : for video, the typical parameters will include picture size, color depth and picture rate. For audio, typical parameters will include sampling rate, bits per sample and loudness. [6]

4.3 System Layer

At this level, QoS parameters are mainly concerned with properties of packets or bits, such as bit rate, packet rate and packet delay. Systems should satisfy these parameters in order to meet the application's requirements. During execution, these parameters must be divided into sub-requirements that must be satisfied by individual subsystems. [6]

5. QoS Negotiation

To negotiate QoS between an application and its underlying system, an application must specify its QoS requirements to the QoS Manager. This is done by the transmission of a set of parameters. Three parameters are of primary interest when it comes to processing and transporting multimedia streams: *bandwidt*h, *latenc*y, and *loss rat*e.

Bandwidth: The bandwidth of a multimedia stream or component is the rate at which data flows through it.

Latency: Latency is the time required for an individual data element to move through a stream from the source to the destination. Of course this may vary depending on the volume of other data in the system and other characteristics of the system load. This variation is termed *jitter* – formally, jitter is the first derivative of the latency.

Loss rate: Since the late delivery of multimedia data is of no value, data elements will be dropped when it is impossible to deliver them before their scheduled delivery time. In a perfectly managed QoS environment, this should never happen. [10]

5.1 Negotiation procedures

In a distributed multimedia applications, the components of a stream are located in several nodes. There is QoS manager at each node. The flow spec is sent to local QoS manager by the source that initiates the flow of data. The manager then checks against its database to see whether the requested resources are available and the QoS can be provided. The flow spec is forwarded to all the nodes where the resources are required to ensure the desired QoS. This flow spec traverses to all the nodes till the destination is reached. At every node the local QoS manager check for the availability of the resources against its database and finally the information that weather the required QoS can be provided or not is passed back to the source. This is the simplest way to negotiate the QoS negotiation. This kind of negotiation is not possible for many purposes, due to conflict between the concurrent negotiations at the different nodes. Applications rarely have fixed QoS requirements. Instead of returning a Boolean value on weather a certain QoS can be provided or not, it is more appropriate to determine what kind of QoS can be provided and let the application determine whether it is acceptable. If the QoS extended is not acceptable the application can degrade to a lower QoS to run itself. If this too is not acceptable the application has to wait for the resources to be freed by other application. It is common to provide a desired and a worst-case value for each QoS parameter. If the QoS requirement changes with the time and the desired quality of the application is not extended by the system it has to renegotiate with the system for the same. [2]

6. QoS Management

When multimedia applications run in networks of personal computers they compete for resources at the workstations running the applications (processor cycles, bus cycles, buffer capacity) and in the networks (physical transmission links, switches, gateways). Workstations and networks may

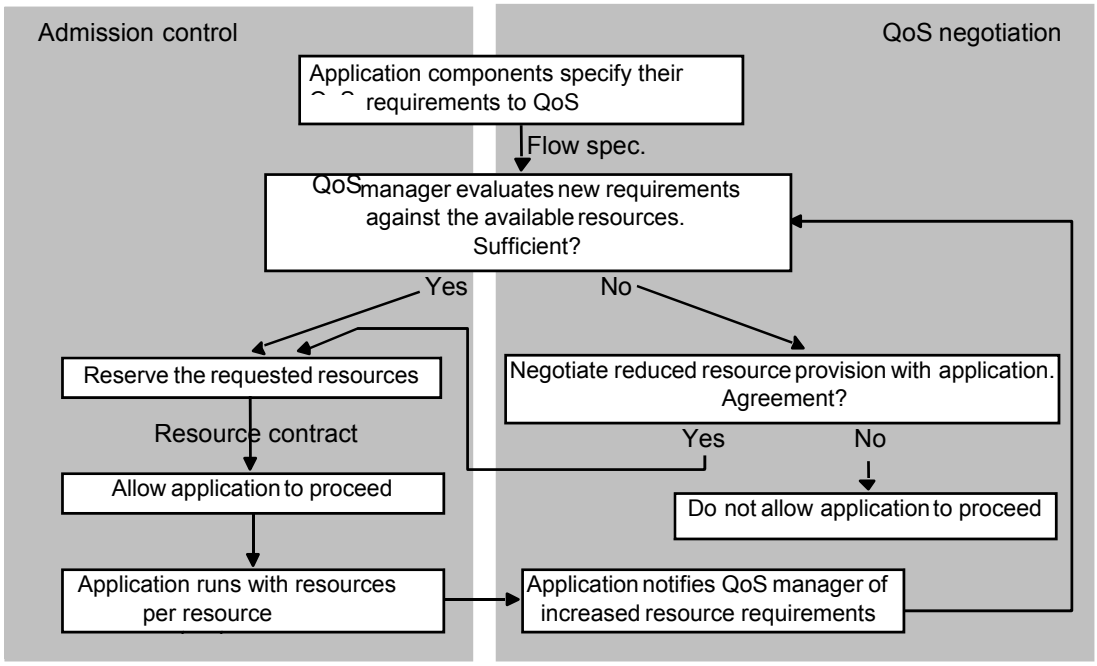
have to support several multimedia and conventional applications. There is competition between the multimedia and conventional applications, between different multimedia applications and even between the media streams within individual applications.[1]

The key feature of these resource allocation schemes is that they handle increases in demand by spreading the available resources more thinly between the competing tasks. Round-robin and other best-efforts methods for sharing processor cycles and network bandwidth cannot meet the needs of multimedia applications. The timely processing and transmission of multimedia streams is crucial for them. Late delivery is valueless. In order to achieve timely delivery, applications need guarantees that the necessary resources will be allocated and scheduled at the required times. The management and allocation of resources to provide such guarantees is referred to as QoS management.

QoS management is defined as the necessary supervision and control to ensure that the desired QoS properties are attained and sustained. [1]

The required resources can only be guaranteed if there is a system component responsible for the allocation and scheduling of those resources.

The figure shows the QoS manager's responsibilities in the form of a flowchart.



The functions of the QoS manager can be classified into two types: [5]

Static QoS management functions

- QoS specification & QoS negotiation
- Admission control & resource reservation

Dynamic QoS management functions

- QoS monitoring, QoS policing

- QoS renegotiation

Some of them have already been discussed earlier. Let us discuss some of the other concepts.

6.1. Admission Control

In general admission control represents the problem of deciding if an application can be supported on a resource. This may include issues of security related to admission, the issues related to capacity and also schedulability on the resource. QoS related to admission control is traffic policing. The system can only provide guarantees as agreed during the QoS negotiation. The function of policing is to ensure that the applications will not generate more traffic than what they specified during the admission test. If application needs to change the QoS specifications, they have to go through the admission control again.

Admission control restricts the number of applications supported on resource. Usually simple admission control decisions are based on worst-case scenario. The worst case policy is based on the observation that the multimedia traffic is characterized by its bur sty nature, therefore, if sufficient resources are not available, then the application may fail. Most of the existing admission control policies are based on greedy strategy. This means that the new application is accepted only if the server could give the client all the requested resources. This can be initiated by the QoS manager as well as by the user. If an application arrives and requests for resources during for its session if the requested resource are available the application is admitted, otherwise, not. This is basis of the greedy approach. An application is accepted as long as the resources are available. The main disadvantage of this approach is if an important application arrives after the resources are already granted to a low priority process, it is rejected (assuming no preemption). Purely predictive strategy, where one trades of the expected benefits in the future with the benefit of the current applications running is more difficult to implement as the future network traffic is unpredictable. A mixture strategy, which combines the advantages of both, is generally suggested and used for admission control. [3]

6.2. QoS Based Resource Management

Resource management at the host and the network level form an important aspect in the guaranteed QoS for the requesting application. Without resource management multimedia systems cannot provide service because transmission over the unmanned resources leads to problems such as dropped packets or delayed packets, violating Quality requirement.

The main goal of a resource management is to provide guaranteed delivery of multimedia data. This management of the specified resources involves three actions

1. To properly allocate resources during the multimedia call establishment so that the traffic may flow according to the specified QoS specification.
2. To control resources during the multimedia data transmission.
3. To adapt to changes to efficiently utilize the existing resources.

6.2.1. Resource Admission

After every layer gets its own QoS specification through negotiation and translation, resource admission based on QoS specification is performed. The resource admission is done using an admission service. This service is embedded in a resource manager. To control resource availability, admission service uses admission tests.

1. Schedulability test for sharing resources e.g. packet schedulability test at the point of admission in the network and each of the node of the network for delay, jitter, throughput and reliability.
2. Spatial tests for buffer allocation for delay and reliability guarantees.
3. Link bandwidth test at the network for throughput guarantees.

6.2.2. Resource Allocation

Based on the results of the admission tests, resource reservation/allocation is performed. The resources are reserved in the path between the sender and receiver. The allocation of resources requires a set of functions embedded in the resource managers and a set of protocols to communicate the information about the allocated resources between the resource managers. The resource managers keep a table of resources allocated and use functions embedded in them to detect and solve conflicts during resource allocation. Allocation of resources can be done in two ways: Pessimistic way or an Optimistic way. The pessimistic approach avoids resource conflicts by making reservations for the worst case. This approach leads to under utilization of resources but results in guaranteed service. In the Optimistic approach the resources are allocated based on average workload. This approach tends to overload resources when unpredictable behavior occurs. The resources are highly utilized but sometimes when the load increases suddenly it may lead to failure. The resource manager solves this by using a function to monitor that detects the overload and preempts application according to their QoS specifications. A resource reservation protocol performs no reservation of resources it is just a channel to communicate the resource requirements and their specified QoS values for various parameters. The messages weather the requested resources are allocated or rejected is communicated using this protocol.

6.2.3 .Resource Deallocation

After the transmission of the data, resources are deallocated, which means, the CPU, network bandwidth, buffer space must be freed and the connection used for the transfer of information has to be closed. This process needs to done without disturbing the other flows in the network. After the close down has been done the resource managers should update the available resource table. In case there is the resource managers to check weather should take a failure in the transmission channel care the resources are still utilized by the applications or not. Proper deallocation of the resources in such a case should be carried in such a case lest the resources will be never used again and leads to underutilization of the available resources. This can be achieved with the help of a monitoring function used by the resource managers that runs periodically checking whether the allocated resources are being used or not.

7. Case Study

7.1 The Tiger Video File Server [1]

An important system component to support consumer oriented multimedia applications is a video storage system that supplies multiple real-time video streams simultaneously. There are several prototypes, which have been developed; The Tiger Video File Server is one of them. This file server system is the one the most advanced systems among them.

7.1.2. Design Goals

7.1.2.1 Video on Demand for Large Number of Users: This system offers a service that is it supplies movies to the paying clients. The movies are selected from the large source of digital

library. As soon as the client issues a request to view a movie this he receives the first few frames of the selected movie within a few seconds then he should be able to perform pause, rewind and fast-forward operations at will. Though the library is a rich source of resources, a few movies, which are very popular, are subjected to multiple unsynchronized requests resulting in several concurrent but time-shifted playing of them.

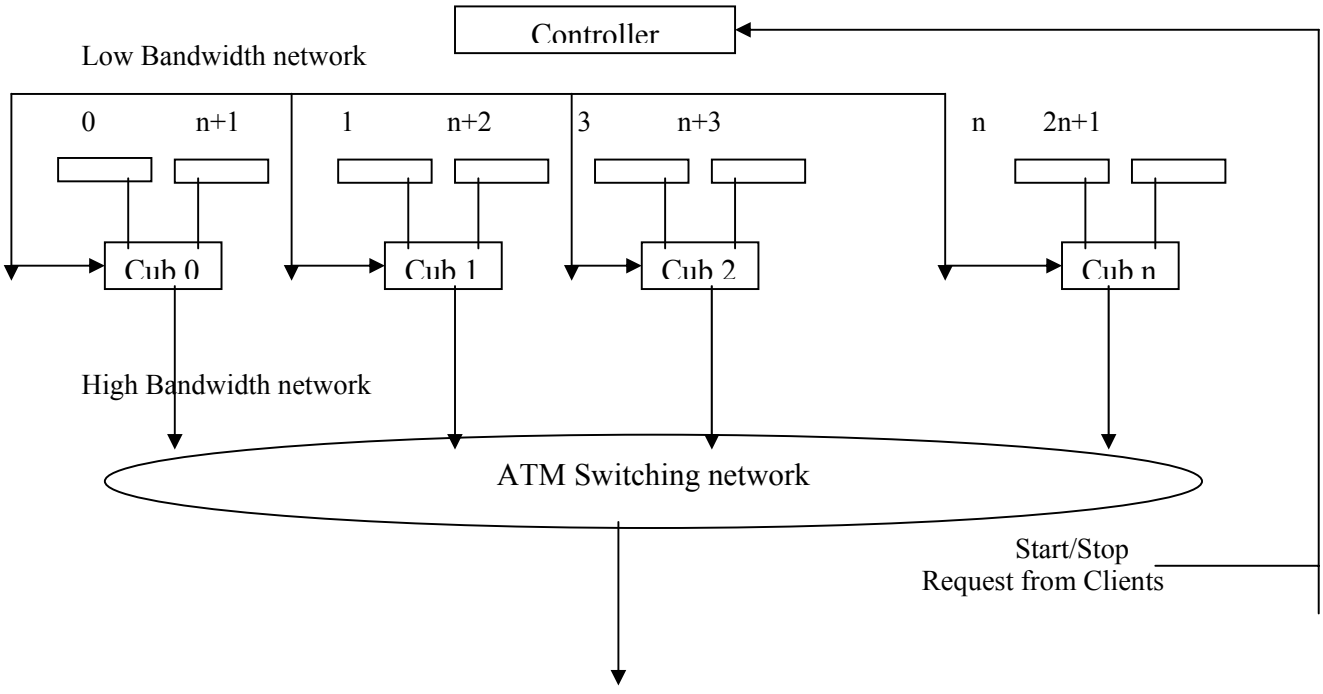
***7.1.2.2 QoS*:** Video streams must be supplied at a constant rate. The data packets received should be in synchronism.

7.1.2.3 Scalable and Distributed: The aim was to design a system with an architecture that can be scalable to support up to 10,000 clients simultaneously.

7.1.2.4 Fault Tolerant: The system should continue to operate without noticeable degradation after the failure of any single server computer or disk drive.

Keeping in view of the above requirements, there is a demand for a radical approach for the storage and retrieval of video data and an effective scheduling algorithm that balances the workload across a large number of similar servers. The primary task is the transfer of high-bandwidth streams of video data from disks storage to a network, and it is this load that has to be shared between the servers.

7.1.3 Architecture: Architecture of a Tiger Hardware is as shown below. The cub computer shown below is equipped with Ethernet and ATM network cards. The Controller is another PC. This is responsible for handling client requests and management of work schedules of cubs.



Tiger Video file Server hardware Configuration

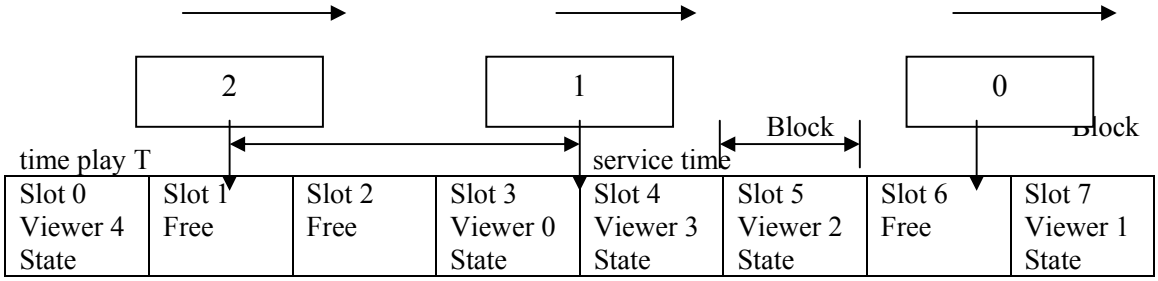
7.1.4 Storage Organization: The key design issue is the distribution of the video data among the disks attached to the cubs in order to enable them to share the load. Since the load may involve

the supply of multiple streams from the same movie as well as from different movies any solution based on the use of a single disk to store each movie is unlikely to achieve this goal. Instead, the movies are stored in a striped representation across all the disks. This leads to the failure of the model because the loss of a disk or a cub results in a gap in the sequence of every movie. The solution to this a storage-mirroring scheme that replicates the data and a fault tolerant mechanism described below

7.1.4.1 Striping: A Movie is divided into set of blocks of around one second each. The set of blocks, which make up the movie, is stored on disks attached to different cubs in the sequence indicated by the disk numbers. A movie can start on any disk. Whenever the highest-numbered disk is encountered, the movie is wrapped around, so that the next block is stored in the disk 0 and the process continues.

7.1.4.2 Mirroring: The mirroring scheme divides each block into several portions, called secondaries. This ensures when the club fails, the extra workload of supplying data for blocks on the failed cub falls on the several remaining cubs, not just one of them.

7.1.5 Distributed Schedule: The workload of the cubs needs to be scheduled. The schedule is organized into a list of slots; each slot represents the work that must be done to play one block of movie. The work involved is reading from the relevant disk and transferring on to network. There is exactly one slot per receiver. Each occupied slot represents one viewer receiving a real-time video data stream. The viewer client in the schedule is represented by (1) The address of the client computer, (2) The identity of the file being played, (3) The viewer's position in the file, (4) The viewer's play sequence number. The schedule is illustrated as shown below.



Tiger Schedule

The block playtime T is the time that will be required for a viewer to display a block on the client computer; it is roughly of about one second. The tiger must maintain interval between the delivery times of the blocks in each stream. Each cub maintains a pointer into the schedule for each disk that it controls and delivery times fall within the current block playtime. The cub steps through the schedule in the real time processing slots as follows (1) Read the next block into buffer storage at the cub. (2) Packetize the block and deliver it into cub's ATM network controller with the address of the client computer.(3) Update the viewer state in the schedule to show the new block and play sequence number and pass the updated slot to the next cub.

7.1.6 Fault Tolerance: The movie files are striped across the entire disk in a tiger system. Failure of any one of the components will result in the disruption of the service to the clients. Tiger offers a remedy to this failure by retrieving the data from the mirrored secondary copies. When a cub or the disk fails an adjacent cub modifies the schedule. As the extra load is shared between other cubs and disks the task continues without disrupting the entire service.

7.1.7 Network Support: The blocks of each movie are simply passed to the ATM network by the cubs that can hold them together with the address of the relevant client. The QoS guarantees of the ATM network protocols are relied upon to deliver the blocks to the client in sequence and time. The client needs sufficient buffer storage to hold two primary blocks, the one that is currently playing the on the client's screen and one that is arriving from the network.

8. Critique and future of QoS in multimedia.

Multimedia applications such as Web surfing, videoconferencing, video-on-demand and multimedia e-mail are becoming increasingly popular in all enterprises. There are obvious advantages in running such applications over existing networks such as the Internet, so that traditional computing applications can be integrated with the newer applications involving person to computer, person to person and multiparty communications. The Internet is already used by almost every individual and organization and is thus an obvious choice for an integrated all purpose network. [11]

Originally, the Internet was designed for data processing applications where delays were relatively
unimportant. A 'best effort' delivery service was adequate in most cases, and in case of loss or corruption of data, the TCP protocol would take care of the necessary retransmission and recovery.

The growth of multimedia applications comprising high quality sound and motion video has introduced new requirements. These applications are bandwidth hungry, and require megabits per second rather than the kilobits per second required for traditional data processing applications.

Another requirement of 'live' (streaming) multimedia applications are timing, both intrastream and interstream. [12] Uncompressed voices for instance cannot tolerate a delay of more than 250 ms before the degradation in quality is noticeable. The same is true for uncompressed video. For motion video, synchronization of sound and picture is essential. More crucial than the actual delay, is the delay variation or jitter. About 10 ms variation in delay can cause problems during playback. This can be overcome to some extent by buffering. Compressed sound and video are even more sensitive to delay variation (1 ms). The best effort philosophy of IP introduces unpredictable delays (and packet losses) across the Internet. The transport protocol (TCP) used to ensure reliability, with its overheads of retransmissions in case of lost or corrupted packets is totally unacceptable. In fact a low level of packet loss may be preferable to additional retransmission delays. Thus 0.1 to 1% of packets (depending on application type, compression scheme) may be dropped without serious degradation of quality [12]

There is thus a need to support a variety of traffic with different quality of service (Q-o-S) requirements. While ATM technology [13] has these features built into the architecture, a global migration from legacy systems to ATM is not likely. The different Q-o-S requirements must be met within the existing TCP/IP architecture. The central issue is how to share available capacity in times of congestion.

Mechanisms are needed
(a) for differentiating between different types of traffic
(priority),
(b) for applications to request network resources (reservation),
(c) ensuring acceptable levels of delay, loss, jitter and throughput (service level agreements).

Research is going on to include the new functionality has to be accommodated in IP routers.

16. Conclusion

End-to-end performance guarantee is required for multimedia communications. This performance can be formally specified by QoS parameters. Different applications have different parameters. Different applications have different QoS requirements. The basic idea of the QoS concept is that the user specifies what he/she wants and the system guarantees the user's requirements if the request is accepted.

References

1. Distributed Systems concepts and design third edition George Coulouris, Jean Dollimore and Tim Kindberg
2. Distribute Multimedia and QoS: a survey Vogel, A.; Kerherve, B; von Bochmann, G.; Gecsei, J. IEEE Multimedia, Volume: 2 Issue: 2 , Summer 1995
3. Distribute Multimedia Systems – Li, V.O.K, Wanjium Liao Commun Sci. Inst., Univ of Southern California, Los Angeles, CA, USA
4. Quality of service management in distributed multimedia systems Guojun Lu Systems, Man and Cybernetics, 1996., IEEE International Conference on , Volume: 2 , 1996
5. QoS-Aware Resource Management for Distributed Multimedia Applications Klara Nahrstedt, Hao-hua Chu and Srinivas Narayan UIUCDCS-R-97-2030
   December 1997
6. Handbook of Multimedia Computing by Borko Furht
7. Quality of Service – IWQoS 2001 – 9th International workshop Karlsruhe, Germany, June 2001 Proceedings
8. IEEE Multimedia Quality of Service Support for Multimedia Applications – Doug Shepherd, Andrew Scott and Tom Rodden, Lancaster University
9. Implementation Issues on Market Based QoS control Hirofumi Yamaki, Yutaka Yamauchi and Toru Ishida. Department of Social Informatiics, Kyoto University
10. IEEE Multimedia. Distributed Multimedia and QoS: A Survey. Andreas Vogel, Brigitte Kerherve and Gregor von Bochmann and Jan Gecsei
11. Multimedia over the Internet: Problems and Solutions. S. McKenzie
12. Sharda N: Multimedia Information Networking, Prentice-Hall 1999, ISBN 0 1325 8773 4, pp 220-220.
13. Halsall, F: Multimedia Communications, Addison-Wesley 2001, ISBN 0 2013 9818 4, pp 646-708.
14. Internet QoS: the Big Picture Xipeng Xiao * and Lionel M. Ni