A Scheme for Dynamic QoS Renegotiation at Intermediate Nodes

*Carlos C. Goulart**
goulart@dpi.ufv.br
Departamento de Informática
Universidade Federal de Viçosa
36570-000 Viçosa–MG

José Marcos S. Nogueira[†]
jmarcos@dcc.ufmg.br
Departamento de Ciência da Computação
Univeersidade Federal de Minas Gerais
31270-010 Belo Horizonte, MG

Gerald W. Neufeld[‡]
neufeld@cs.ubc.ca
Department of Computer Science
University of British Columbia
Vancouver, B.C., V6T 1Z4
Canada

Resumo

Esse artigo apresenta um esquema para a renegociação dinâmica da Qualidade de Serviço (QoS) de aplicações multimídia distribuídas em nós intermediários. O objetivo principal da renegociação dinâmica é explorar o comportamento dinâmico das conexões multimídia para maximizar a utilização de recursos e, como consequência, acomodar um número maior de conexões. O conceito de QoS mínima para cada conexão é usado como mecanismo para reserva de recursos nos nós intermediários. Este mecanismo dá uma prioridade implícita maior para os pedidos de renegociação do que para pedido de novas conexões. Algumas simulações são apresentadas mostrando que o modelo pode aumentar o número de conexões admitidas e, ao mesmo tempo, manter a taxa de rejeições de pedidos de renegociação em níveis bem baixos.

Abstract

This paper presents a scheme for dynamic Quality of Service (QoS) renegotiation for distributed multimedia applications at intermediate nodes. The major goal of dynamic renegotiation is to explore the dynamic behavior of multimedia connections to maximize resource utilization, and as a consequence, accommodate a greater number of connections. The concept of minimum QoS for each connection is used as the mechanism for reserving resources at intermediate nodes. This mechanism gives an implicit higher priority to renegotiation requests over new connection requests. Some simulations of the proposed model are presented and their results show that it is possible to increase the number of active connections, and to keep renegotiation rejections to a very low level.

1 Introduction

The emergence and fast growth of high speed networks has led to development of new kind of applications and new communication protocol paradigms to support such applications in a more efficient manner. An important new class of applications is the class of multimedia applications. Multimedia applications comprise different types of information (e.g. video, audio and text) that are carried over the same physical link, and so, they require very high bandwidth to be supported. Moreover, multimedia applications require synchronization among these different types of information. Therefore, the protocol stack that supports multimedia applications has to consider these characteristics (bandwidth and synchronization) in order to

*Ph.D candidate at DCC/UFMG (supported by CAPES-PICD program) and Assistant Professor at DPI/UFV
†Associate Professor at DCC/UFMG
‡Associate professor at DCS/UBC

provide an efficient communication facility. The two best known protocols stacks, Internet and OSI, do not currently address these issues[Bla92, Tou95]. Many proposals[Cam93, Cou95, Fer92, Haf95, Liu93] have been developed to try to solve this problem. All these proposals use the concept of Quality of Service.

Quality of Service (QoS) is a general way to express a set of characteristics for a multimedia application. At the highest level, QoS can be interpreted as an image quality (e.g. color or black-and-white) or a sound quality (e.g. telephone or CD quality). At lower levels, QoS has to be mapped to more concrete parameters such as bits per second or transit delay. Therefore, QoS is required through the entire system. Defining the QoS for a distributed multimedia application is a complex task that can involve many components in the system.

In order to provide a desired QoS level to a specific multimedia connection, some amount of resources is reserved to it. These resources can be of different types like bandwidth, CPU cycles, buffer space, etc. Due to the dynamic aspect of multimedia connections, it would be desirable that the status of reserved resources could change along the connection life time. Therefore, if a connection is no longer needing a large amount of resource, it could release a portion that could be used by other connections.

From an application's point of view, it is important to determine the end-to-end QoS, which means that resources should be reserved throughout the connection's path. In this case, each node along the path must support resource reservation in order to provide the desired end-to-end QoS level. Thus, node behavior is a key point for assuring end-to-end QoS. This paper presents an analysis of node behavior supporting multimedia connections and using resource reservation.

This paper is presented according to the following organization: section 2 presents an overview of multimedia applications and quality of service; a proposed model for dynamic renegotiation of QoS is presented in section 3, as well as the node architecture to support it and some considerations about feasibility of the model; in section 4, a simulation of the proposed model is presented and some aspects related to its implementation are described, as well as its results; and, finally, section 6 contains the conclusions and future work.

2 Multimedia Applications and Quality of Service

Multimedia applications are composed by different media, and each media has its own characteristics and requirements. Some media must be continuous from a user's point of view (e.g. video and audio) while others can be discrete (e.g. text, single images). Some media tolerate a small error rate while others have to be error-free. From this diversity of characteristics comes the need to describe the requirements for each individual media that comprises a specific multimedia application. The Quality of Service (QoS) parameters are used to perform such a task.

However, the exact meaning of QoS is not completely well defined. There are some different and fuzzy definitions related to QoS and some different QoS meanings depending on the point of view. For distributed multimedia applications one of these definitions states that: "Quality of Service represents the set of those quantitative and qualitative characteristics of a distributed multimedia system necessary to achieve the required functionality of an application"[Vog95]. As the definition implies, QoS can range from a user-defined subjective parameter, for instance the quality of an image, to a precise parameter on the transmission system as the transit delay. Therefore, in a distributed multimedia system, the QoS processing involves many different, but inter-related tasks that can be summarized into three major steps[Vog95]:

- Estimate the QoS requirements from a user's point of view. This step determines the subjective wishes of the user concerning to the quality of the service and may include performance, synchronization, cost and so forth.
- Map the estimates onto QoS parameters for all the system components involved in the multimedia

application. Here, the user's wishes must be translated into real parameters like throughput, transit delay, delay variation (jitter), error rate, and so forth. In this phase the communication sub-system characteristics must be considered.

- Negotiate with the system components to achieve the desired QoS. This negotiation involves local and remote negotiation, as well as a distributed negotiation for gathering resources from the communication sub-system at intermediate nodes.

The definition of QoS parameters depends on the communication protocol stack that is being used. If the protocols have some support for QoS definition they would ease this task. There are three levels of QoS defined according to the communication protocols: Application layer, Network/Transport layer and Lower layers[Zha93]. After defining QoS requirements and mapping them, the users and providers of a multimedia application can establish an agreement on what kind of QoS parameters will be used for a specific session. The application can be started only if negotiation leads to an agreement. There are different types of agreement that include guaranteed, predictive, or even best-effort. The greater the guarantee level the more resources must be reserved to a particular connection.

Despite the agreement, QoS must be monitored during the application life time to ensure that the contract will not be broken, intentionally or not. An initial agreement does not mean that the application's users will keep the same requirements forever. The QoS could change during an application session. These changes on QoS level imply that more resources must be requested from the supporting communication system or even that some resources can be released. In fact, the QoS renegotiation is a fundamental activity to be performed by systems supporting multimedia applications.

Another important aspect of multimedia applications is related to time constraints, which require a real time characteristic for some applications. There are some different classifications for real time multimedia applications based on their time constraints. One such classification is presented in[Tou95]. The general idea behind the real time concept is that the final user should have the impression that an application is being performed in real time and any interaction should be answered instantly. Therefore, the overhead caused by introducing QoS renegotiation must be kept small enough in order not to interfere with application's real time characteristics.

Resources in distributed systems are managed by various components of a resource management system. For a multimedia distributed system, Quality of Service management is a key activity to be performed. Quality of Service management can be grouped into the following major activities:

1. provide the initial set of resources for establishing a multimedia session, according to the QoS specification;

2. monitor the traffic and apply some control policy to avoid QoS violation;

3. change the resource set to reflect new QoS requirements during the multimedia session.

In order to provide these activities some QoS management functions are already defined. These functions include: QoS specification and mapping as discussed previously; *QoS negotiation* and *resource reservation* which define the initial set of resources to support the desired QoS; *QoS monitoring* and *source policing* to avoid or minimize QoS violations; *QoS adaptation* to adapt QoS parameters to reflect a new environment's status; *QoS renegotiation* to change QoS parameters under user request; *QoS accounting* that is responsible for determining the cost associated with each QoS level; and *QoS termination* that has to free all resources allocated to a specific connection. As QoS adaptation and QoS renegotiation are close related to the proposal presented in this paper, their definition will be highlighted. QoS adaptation is responsible for exhibiting a graceful degradation in case of changes in the environment (e.g. congestion). It should be performed as transparent as possible from user's perspective. QoS renegotiation function permits changing QoS parameters during a session. The user has to request a renegotiation to change QoS parameters previously negotiated.

There are various proposals to deal with the QoS negotiation and most of them highlight the need of QoS renegotiation[Cam93, Nah95, IETF96]. The scheme described in this paper tries to show a way of implementing the dynamic renegotiation concept. The scheme is general enough to be used with OSI or Internet protocol and it utilizes some basic concepts that can be mapped into both models (e.g. MIB - Management Information Base). Our major focus is placed on negotiations at intermediate nodes where resources are shared by many connections. The present scheme is intended to work together with some traffic control mechanism at the interface between service users (or providers) and the network as defined in[ATM95].

3 Proposed Model for Dynamic QoS Renegotiation

The major motivation for dynamic QoS negotiation is to explore the dynamic behavior of multimedia connections in order to improve resource utilization and, as a consequence, increase the number of connections supported by a node. Consider, as an example, a video connection with a variable bit rate as shown in Figure 1. If the Peak Rate were reserved for the connection that will support this exhibition, most of the bandwidth would be wasted.

If the QoS management could predict this pattern, it could change the QoS in advance to support the traffic, as shown in Figure 1.b. For some kind of application, like video on demand, it is a very reasonable assumption, since the video sequences are coded and stored off-line. The codification scheme can provide information about bit rate variation along the time. Other information necessary for updating the amount of resources needed is the maximum delay between user and server. A maximum bound for delay values can be computed even for networks that use non-fixed size packets[Ram92]. Using this information it is possible to optmize the scheduling of renegotiation requests[Gro95].

For applications in which the future bit rate is not known (e.g. videoconferencing) this scheme can also be used. In this case the QoS renegotiation scheme will determine a maximum time interval to change the QoS. Depending on the delay value some data loss can occur when there is a change from a lower bit rate to a higher one, at the beginning of higher rate interval. For instance, in a videoconferencing the video information can change its bit rate when someone either moves or shows a picture in front of the camera. If the distances are too long, some of the first frames could be lost, but the rest of the frames could still be received adequately. Based on the delay value the user application and the QoS manager can decide either to use dynamic QoS renegotiation or to reserve resources based on the application's peak rate. Of course both options will not have the same cost. It is shown in [Gro95] that a renegotiation scheme is also useful for saving buffer space at the receiver side.

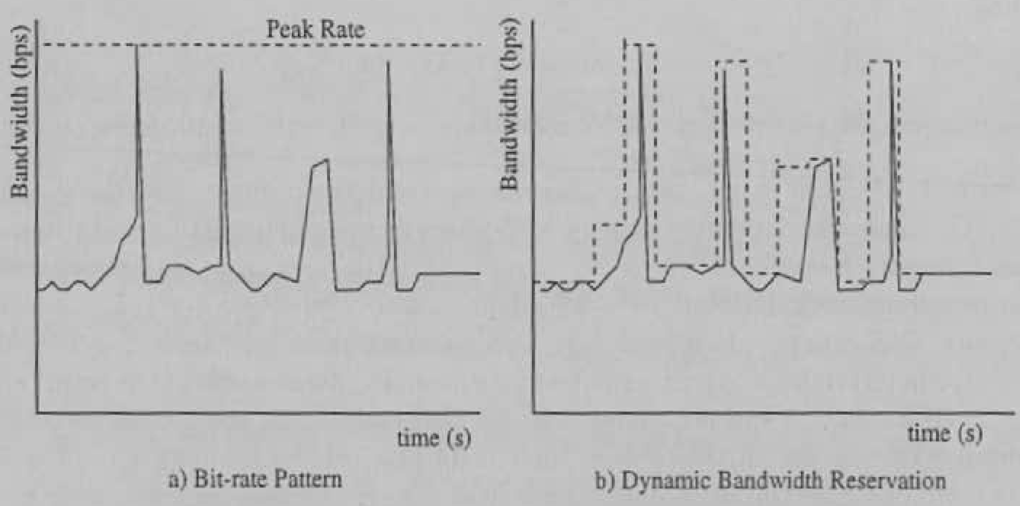

Figure 1: Variable Bit Rate (VBR) Pattern

Each request for a new QoS level should be received and submitted to a connection admission control (CAC) module to verify if there are sufficient available resources, as if it were a new connection request[IETF96]. A request for a lower level of QoS must never be rejected, since it would not need extra resources. In case of a greater level request, a rejection could occur and this result can cause some impact on the application. In our approach, each connection should inform a minimum value to its QoS and this value will be reserved. Thus, in case of renegotiation rejection the QoS level in use will remain, and it will be greater than or equal to the minimum level, which is supposed to support the application at an acceptable quality.

The major idea behind this proposal is to *combine QoS adaptation and QoS renegotiation functions to provide a new function called dynamic renegotiation*. A dynamic renegotiation will be requested by a QoS manager and in this aspect it is similar to renegotiation function. But dynamic renegotiation has to be performed transparently from the user's perspective, as QoS adaptation function. Besides combining these two concepts, the proposed model uses the idea of minimum QoS for providing resource reservation to accepted connections. In the next section the model for dynamic renegotiation will be detailed.

3.1 Minimum QoS Reservation

In the proposed model a minimum QoS requirement for each connection is used to reserve resources. Each connection request must specify two values for each QoS parameter: a minimum level which is used to reserve resources for a specific connection, and; an initial value which is used to allocate initial resources for an accepted connection.

The connection admission control algorithm will utilize these two values to either accept or reject the new connection, as well as to reserve resources in case of acceptance. The basic criteria to accept (or reject) a new connection is defined by the following comparison:

```
if (total_resources - allocated_resources) >= initial_request
    accept_connection;
else reject_connection;
```

where: $allocated\_resources =$

$$\sum_{i=1}^{n} \begin{cases} Minimum\_resource_i, & if current\_resource_i < Minimum\_resource_i \\ current\_resource_i & otherwise \end{cases}$$

total_resources is the total amount of a resource type on a specific node and n is the number of active connections.

To accept or reject a renegotiation request the comparison is made using the *current resources* instead of the *allocated resources*. *Current resources* is the amount of resource that is really being used at any instant in time. As will be seen in our simulation (see section 4) there are three allocation methods that can be considered: Peak, Average and Dynamic. For each allocation method the value of *allocated bandwidth* will have different values while the value of *current* bandwidth can be the same for all three allocation methods (where all three methods have accepted the same set of connections). The criteria to accept or reject a new (and greater) resource level is defined by the following comparison:

```
if (total_resources - current_resources) >= requested_resources
    accept_renegotiation;
else reject_renegotiation;
```

where:

$$current\_resources = \sum_{i=1}^{n} current\_resources_i$$

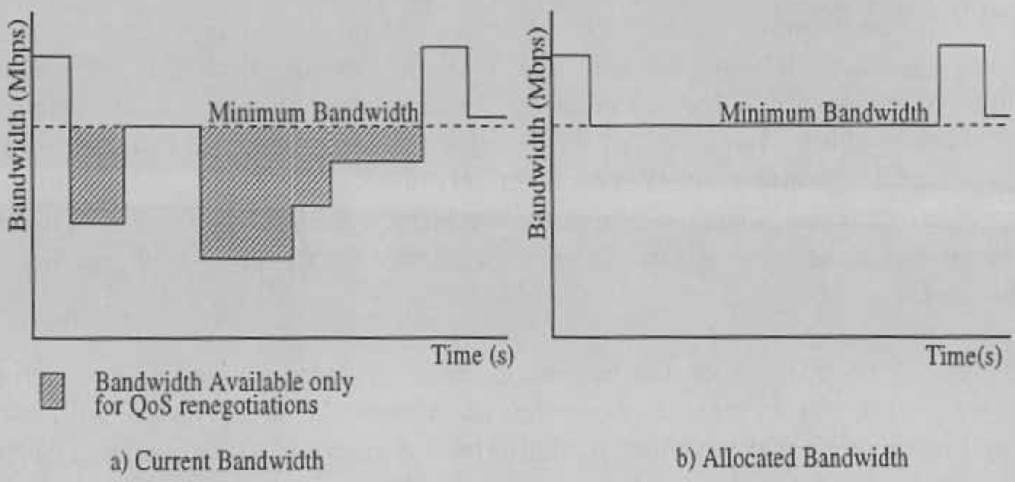

Figure 2: Minimum QoS Reservation

As an example, consider the bandwidth as one of the resources that is being allocated. In this case, minimum QoS corresponds to a *minimum bandwidth* level. The minimum bandwidth level works as a resource reservation scheme for each connection. When the amount of bandwidth needed by a specific connection is greater than its minimum, the *allocated bandwidth* is equal to the *current bandwidth.* If the value of bandwidth needed is smaller than minimum value, the *allocated bandwidth* will be equal to the *minimum* value, as shown in Figures 2.a and 2.b.

The use of the idea of minimum QoS level for each connection will increase the chances of accepting a new level of QoS. Both new connection requests and new QoS level requests depend on resource availability, but the concept of available resource is different for each one. *Allocated bandwidth* is considered for accepting new connections while *current bandwidth* is used for accepting renegotiation requests. In other words, a renegotiation request will have a kind of priority over new connection request since an amount of resources could be exclusively used by renegotiation requests. This situation can also be seen in Figure 2.a. Besides that, this kind of reservation will limit the number of connection to be accepted, which will be translated into a high guarantee level for the accepted connections.

3.2 Architecture for Dynamic QoS Renegotiation

Figure 3 shows a possible architecture for implementing the QoS negotiation. This architecture can be used for initial QoS negotiation as well as for dynamic QoS negotiation. The QoS manager is responsible for obtaining options from the user application and for translating them to real QoS parameters. After this, the QoS manager initiates a local negotiation to reserve local resources to support the application. If local resources are available, the QoS manager issues a request for resources from the network. QoS manager is also responsible for issuing renegotiation requests during session time.

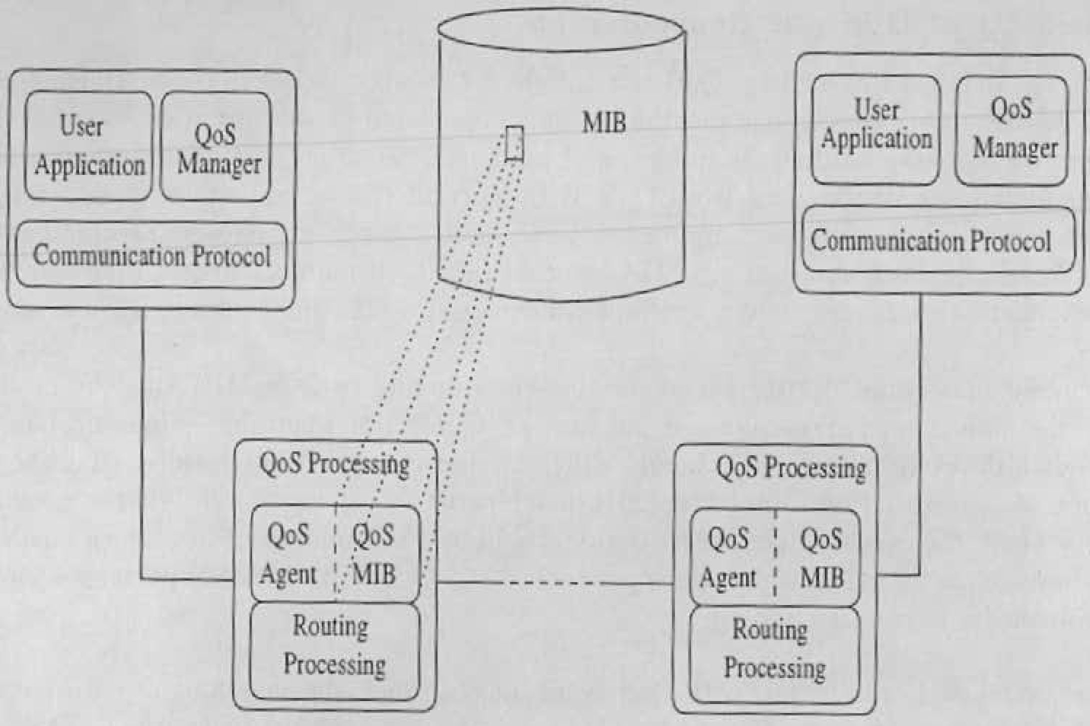

Figure 3: QoS Negotiation Architecture

QoS MIB (Management Information Base) has the same meaning as the regular MIB (Management Information Base)[RFC1213]. But since it deals with connections supporting applications with real time characteristics this MIB has some special features. Each connection has just a few parameters placed into the QoS MIB, only the parameters necessary to identify the connection and the parameters involved in the QoS renegotiation such as bandwidth and delay. Since QoS processing has to be performed within severe time constraints, QoS MIB has to be stored on main memory. Therefore, QoS MIB represents a small subset of the real MIB that is stored on disk.

QoS Agents must be located at every node of the network to perform local computation and to decide about resource availability. The QoS Agent gets the request issued by the QoS Manager and checks if the node has sufficient resources to support the connection. In case it does, the request is forwarded to the next node in the path and a temporary resource reservation is done. If any node can not support the connection, it sends backward a cancellation. If the request arrives at the destination, and resource reservation is also successful there, then a confirmation is sent through the established path and the service can be started with the desired QoS level. This basic algorithm for initial QoS negotiation can be improved to increase the chances of success or to give a better feedback information in case of rejection[Ram92].

It should be noted that the initial negotiation happens before the service starts involving user interaction for selecting parameters. To improve this initial QoS negotiation the user can define profiles which contain preferences for each type of service (video on demand, teleconferencing, etc). However, after starting a service any QoS renegotiation must be done in a real-time basis. It means that any change is bounded to a maximum period of time. Dynamic QoS negotiation differs from the conventional management functions due to its real time characteristic. If someone, for instance, requests a better video quality, (s)he would like the response as soon as possible. For connections with Variable Bit Rate (VBR) pattern, QoS changes could be so frequent that they should be handled transparently to users. For that reason, the task is left to the QoS management system. The impact introduced by renegotiation processing should not interfere with application's real time characteristic.

3.3 Feasibility of Dynamic Renegotiation

To estimate the overhead caused by QoS renegotiation two approaches must be considered: the node capacity to perform QoS processing and the traffic increase due to renegotiation requests. To estimate the first one, we can assume that the following operations have to be performed in order to change the connection's QoS parameters: read from QoS MIB (current connection's QoS parameters and node's available resources), perform some computation using these two set of values and write them back to the QoS MIB. We can make a non optimistic assumption that each memory access takes 4 machine cycles and the computation consumes 1000 operations which are two machine cycles long on average.

As discussed in section 3.2 the set of parameter belonging to QoS MIB must be minimal. In our simulation, the number of parameters is equal to four (connection identifier, minimum bandwidth level, current bandwidth level, and available bandwidth). To perform the computations using these four parameters, the QoS renegotiation would take 2016 machine cycles at each node. If the machine works at 200MHz clock rate, the overhead per renegotiation would be 10.08 microseconds, which means that a node can handle more than 99,000 renegotiations per second. Consequently, node capacity for QoS processing is not a problem for current technology.

However nodes do not only deal with QoS processing. In fact, the switching of information packets is still the major function of intermediate nodes. Management packets must be kept below 1% within a total traffic[ATM95]. Considering an intermediate node with total processing capacity of *Cap* bits per second that accept connections with peak rate *Peak*, this node could accept up to $Cap/Peak$ connections using peak allocation. The total traffic generated by renegotiation requests arriving at this node will be equal to $Num\_conn \times Ren\_rate \times Size$, where *Num_conn* is the number of connections, *Ren_rate* is the average renegotiation rate per connection and *Size* is the size (in bits) of a renegotiation packet. Assuming that dynamic allocation could double the number of accepted connections and that traffic generated by renegotiation requests must be below 0.5% (up to half of all management packets), the average renegotiation rate per connection should be limited by the following inequality:

$$Num\_conn \times Ren\_rate \times Size < 0.005Cap, \textit{ or}$$

$$Ren\_rat < 0.0025 \times \frac{Peak}{Size}$$

Considering peak rate equal to 6 Mbps and packet size equal to 64 bytes, this will represent an average value up to 29 renegotiations/second allowed to each connection. This value can accommodate renegotiation scheme since coded video is expected to issue just a few renegotiations per second[Gro95]. The size of the renegotiation packet is small, since it has to carry just the essential information related to the new QoS level that is being requested, as well as adressing information. The size of the renegotiation packets are not related to those of the data packets.

These results show that QoS renegotiation can be used to effectively improve the network resource utilization. Some machine features such as, cache memory, associative memory, faster clock rates and so forth can help in keeping the QoS renegotiation as low as possible. Besides that, QoS processing can be performed in parallel with the routing processing since there are two independent processing modules in the proposed architecture (see Figure 3).

4 Model Simulation

4.1 Overview of Simulation Problem

The objective of the simulation is to verify the behavior of a specific node subject to admission control based on the proposed model. The set of events that a node can handle comprises: connection establishment request, two types of renegotiation requests (either for getting more resources or for releasing resources), cancelation of a connection establishment request, cancelation of a renegotiation request and connection termination. For each event the node must take some actions that can be summarized in reserving or releasing some resources. A node has a finite amount of resources that must be shared among all accepted connections. Thus, the basic criteria to accept a new connection or a higher QoS level is determined by resource availability. In this approach, bandwidth is the only type of resource that is being considered.

Three different bandwidth allocation methods can be used: Peak, Average and Dynamic. In the Peak allocation, for each accepted connection an amount of bandwidth equal to its peak rate will be allocated. In this case, an 100% guarantee is achieved but a great amount of bandwidth can be wasted. In the Average allocation, the average rate is allocated to each accepted connection. Dynamic allocation will allocate some bandwidth amount just for the next few seconds and request a new allocation periodically, with a non-fixed period of time. The simulation uses all three allocation schemes. In the simulation, a specific connection could be accepted by all three schemes, or by average and dynamic schemes, or yet by either dynamic or average.

Some simulations hypotheses are assumed:

1. there will be a great number of connection of the same type (VBR connections);
2. connection establishment requests are uniformly distributed along the simulation time;
3. the duration of connections vary from a few minutes (e.g short video clip) to 2 hours (e.g. movie);
4. the connection's bit rate is limited to a range from 1.5 to 6 Mbps and bit rate levels remain the same during a period varying from 1 to 10 seconds;
5. bit rate pattern for each connection is known by the receiver side in advance.

The first hypothesis is important since a statistical reservation scheme is adopted in our approach. However, if there are few connections the node shall not be overloaded and there will be no problem of lack of resources. The second hypothesis gives a dynamic aspect to simulation which can have new connection requests during all simulation period. The fourth hypothesis is related to the behavior of coded VBR connections that can generate bursts as long as tens of seconds[Gro95] and it is also related to the interval of scene changes that can vary according to the type of video (e.g. quiz show, sports, an so on)[Dag94].

The last hypothesis is not provided by codification scheme, but it can be generated at codification time in a separate data structure. This data structure will not have an important impact on the amount of data to be transmitted. For instance, a 2-hour long video will generate up to 7200 renegotiation requests (according to fourth hypothesis) and each request will have to contain information of the new requested value and timestamp. Renegotiations must be initiated by the receiver side in order to conform with multicasting transmissions in which renegotiation scheme could stop before arriving to the server due to use of filtering and merging techniques[Zha93]. For the sake of simplicity, in this simulation just discrete values of 0.5 Mbps steps are considered for bit rate.

A set of C programs has been developed to perform the simulation task, and each of them is composed by three major modules:

- a simple user interface to request some simulation parameters (simulation time, total node bandwidth, number of connections, etc.);
- a random number generator module that is responsible for defining the list of events (connection request, connection termination, renegotiation request) and the connection VBR behavior.
- an accounting module that generates statistic results.

In its first step, the program generates a list of connection requests and puts them into a list of events. This list is kept ordered since every new element is inserted according to its timestamp. The rest of the program behavior consists in submitting events to a node and computing values for allocated bandwidth, current bandwidth, number of active connections, number of accepted (rejected) connections, number of accepted (rejected) renegotiations, etc.

Simulation is divided into two parts. In the first part a single node is simulated and its major goal is to verify how effective is the model to acommodate a greater number of connections keeping renegotiation rejections at a low level. Some comparisons between the number of conections allocated by Peak allocation and by Dynamic allocation are considered. Bandwidth utilization by these two allocation methods are also shown. Two different models for a node are considered. Firstly, it is considered a simpler model in which there is only a single channel with capacity equal to the node capacity. In the second approach the node has four channels and each channel has one-quarter of node's capacity.

In the second part a network of seven nodes is simulated to identify the behavior of nodes with lower processing capacity and to verify renegotiation rejection rates in the network as a whole. This part of simulation contains all the events of renegotiation protocol. In the first part events such as *connection_cancel* and *bandwidth_cancel* are not present since there was only one node. Now, with seven nodes a path is established for each connection. After receiving an event, the node performs some computation and issue an event either to the next node in the path, in case of success, or back to the previous node in the path, otherwise. Table 1 summarizes the events received and issued by an intermediated node.

Input event	Result	Output event	Direction
connection_request	accepted	connection_request	next_node
connection_request	rejected	connection_cancel	previous_node
reserve_bandwidth	accepted	reserve_bandwidth	next_node
reserve_bandwidth	rejected	bandwidth_cancel	previous_node
release_bandwidth	accepted*	release_bandwidth	next_node
bandwidth_cancel	accepted*	bandwidth_cancel	previous_node
connection_cancel	accepted*	connection_cancel	previous_node
connection_termination	accepted*	connection_termination	next_node

(*) events that are always accepted

Table 1 - Input and Output events of simulation

4.2 Simulation Results

All results described in this section were obtained using VBR connections with peak rate equal to 6 Mbps and simulation time equal to 15 minutes. The value for the peak rate was chosen according to the MPEG-II standard for a video signal with NTSC-like quality. The simulation time was chosen considering two aspects: (1) it should be long enough to permit an observation of a high workload; and (2) it should be bounded to a maximum value in order not to increase the computation time. The definition of a maximum simulation time is possible in this case since the behavior after beginning the high workload period does not exhibit significative variations.

Other simulation parameters like node capacity, link capacity and number of conections vary according to simulation type. In all simulations these parameters were chosen to characterize a high workload: a great number of connections in a short period of time. Most of the connections had duration time greater than simulation time. In this case, the simulation program generates a great number of connection termination at the end of simulation interval, that are not shown in the graphics of this section.

4.2.1 Single Node

The first simulation had the purpose of measuring the behavior of bandwidth utilization for the three allocation methods. In this case, the node capacity is equal to 600 Mbps and there is no reservation of minimum bandwidth. As resource reservation is not used, peak allocation has the worst result from resource utilization's point of view. Other expected result, that has been confirmed, is that accepting more connections using dynamic or average allocation can lead to a very high rejection rate for renegotiation request (around 18%), since the node accepts more connections than it can handle.

As can be seen in Figure 4, in the beginning, when there is enough available resources, connection requests are accepted by all three allocation methods and resource utilization is the same for them. But they have different saturation points. We refer to saturation as the maximum number of connections that a node can accept during a time interval, according to a specific allocation method.

This previous result shows that it is imperative to define a scheme that minimize the probability of renegotiation rejections. In other words, a connection must be accepted only if there will be enough resources to transport its data according to a specified guarantee level, for the entire connection time. As explained in the previous section, the idea of minimum bandwidth per connection is used. Each connection request specifies a minimum value to its bandwidth and this value is defined as a minimum threshold for its allocated bandwidth. The maximum value is a fixed value specified within a node that applies to all connections. If the minimum value specified is greater than or equal to peak rate then the connection will have 100% guarantee.

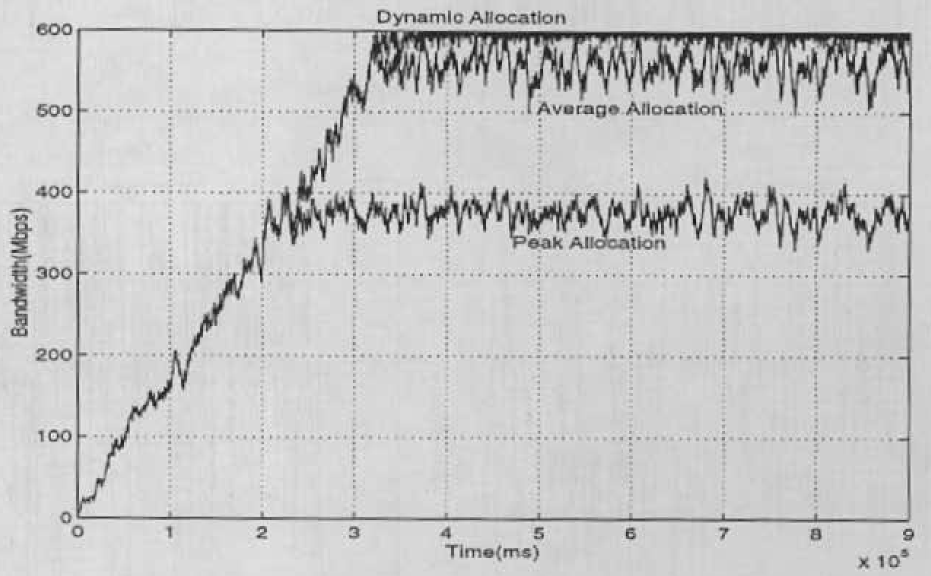

Figure 4: Evolution of Bandwidth Utilization

Of course, these levels of guarantee are just theoretical since all connections are sharing the same amount of bandwidth. In other words, there is not per connection reservation in this simulation. This could seem an unfair scheme since a connection that had reserved its peak rate could have some of its packets discarded. A priority scheme can be used to solve this problem and increase the guarantee level based on the amount of reserved resources. The more bandwidth a connection reserves, the higher will be its priority. In case of congestion, packets of low priority would be discarded first. In extreme cases, where all connections reserve either nothing or their peak, the scheme will correspond to best-effort or peak allocation, respectivelly. But it is assumed that a connection will normally reserve a minimum amount of bandwidth and that this amount should be sufficient to support the service with minimal quality.

In this simulation each connection specifies its minimum bandwidth requirement ranging from 1.5 Mbps to a *maximum value*. When the *maximum value* allowed for the *minimum reservation* is set to the peak rate (6 Mbps), it can be observed that the total bandwidth utilization remains below 90% (see upper-left graphic on Figure 5). This is due to the contribution of connections that had reserved peak rate (or around it). This means that more than 60 Mbps is never used, which could acommodate at least 10 more connections reserving their peak rate. In this case there are no renegotiation rejections. When the *maximum value* for bandwidth reserved is fixed at 3Mbps (*minimum bandwidth* can vary from 0 to 3 Mbps, but peak rate is still equal to 6 Mbps), resource utilization increases and some renegotiation rejections occurred. These two cases are presented in Figure 5 together with two intermediate cases (*maximum value* for *minimum reservation* equal to 4 and 5 Mbps). All graphics in Figure 5 were generated only for periods when the node is saturated, which used to start around 4.5 minutes and remains until the end of simulation time (15 minutes).

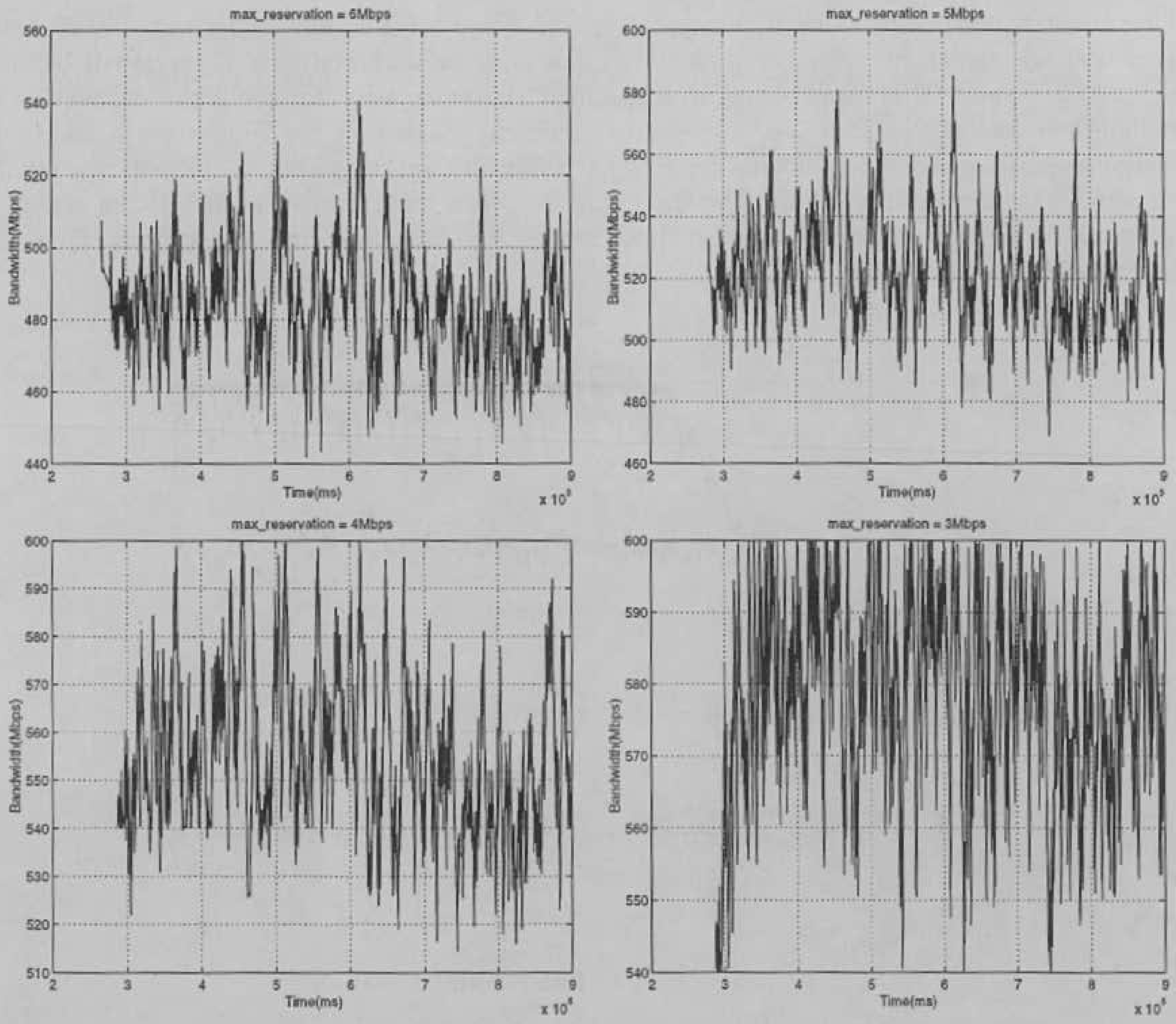

Figure 5: Bandwidth Utilization Patterns

Table 2 summarizes the results for six values of the maximum allowed bandwidth reservation. Average bandwidth utilization values were computed only for periods of saturation. The last column on Table 2 represents the maximum number of active connections compared to the maximum number of connections allowed for Peak allocation (100 peak allocated connections where each connection reserves 6 Mbps). This represents a service improvement up to 46% with a very low renegotiation rate provided by dynamic negotiation.

Reservation Range	Bandwidth Utiliz.(%)		Reneg. Rejec. (%)	Gain over Peak Alloc.(%)
	Average	Peak		
none-6.0	70.70	89.40	0.000	31.0
none-5.5	83.56	94.18	0.000	38.8
none-5.0	86.96	95.83	0.000	42.2
none-4.5	89.58	100.0	0.005	46.2
none-4.0	92.72	100.0	0.128	52.2
none-3.5	94.81	100.0	0.449	55.4

Table 2 - Bandwidth Utilization in Function of Bandwidth Reservation

The model of node used to obtain these first results is very simple: the node has a single output channel with capacity equal to the node's capacity. If the node's architecture is modified and the node capacity is divided into n output channels of the same capacity the results are a bit different. This difference is due to the influence of channel capacity on connection acceptance and renegotiation rejections. Besides that, bandwidth utilization is lower since the maximum at each channel do not happen at the same time. Figure 6 shows a comparison among some results for renegotiation rejection rate for nodes with 4 output channels with 150 Mbps and 300 Mbps, respectively.

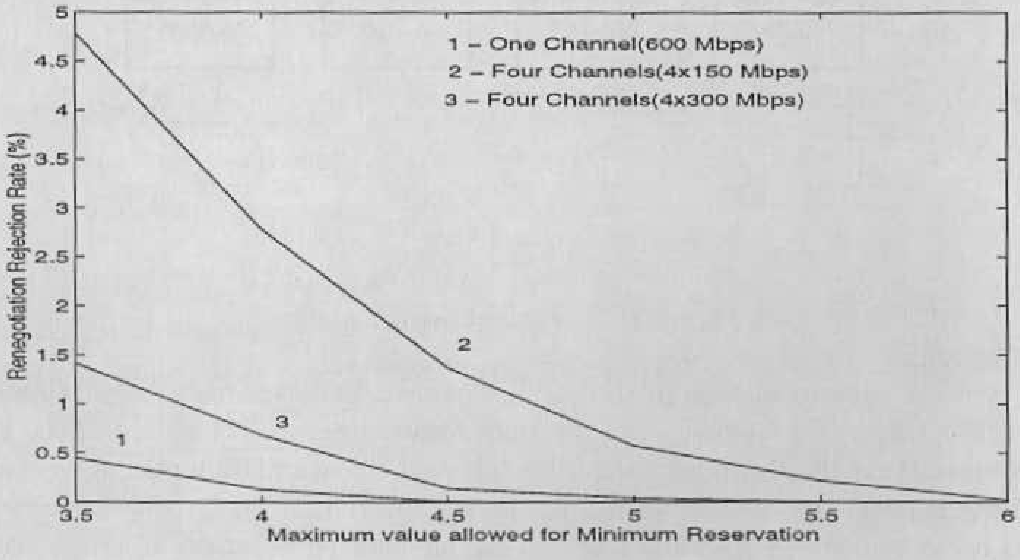

Figure 6: Comparison of Renegotiation Rates

4.2.2 A Small Network

In the second part of simulation a network with seven nodes is organized following a tree topology, as shown in Figure 7 is considered. The major goal in this part is to simulate the complete protocol for admitting connections and reserving resources, since with a single node some events are not present, as connection establishment cancelation.

One typical application that can be supported by this topology is video-on-demand, where the video server is located at node 0, and the users are connected through nodes 3, 4, 5 and 6. For this part we have defined two different types of networks, where each one has a parameter n defined as the basic link capacity:

1. balanced network: where the processing capacity of each node is equal to the sum of capacities of all its input[1] links. Processing capacities are equal to $2n$, $4n$, and $8n$, for nodes at levels 2, 1 and 0, respectively. Link capacities are equal to n for links between levels 2 and 1, equal to $4n$ for

links between levels 1 and 0 and equal to $8n$ for the link connecting the video server. In this case bottleneck nodes are located at nodes on the level 2.

2. unbalanced network: where the processing capacity of each node is lower than the sum of capacities of all its input links. Processing capacities are $2n$ for all nodes. Link capacities are n for links between levels 2 and 1, and $2n$ for links between levels 1 and 0 and for the link connecting the video server. In this case bottleneck node is the node 0.

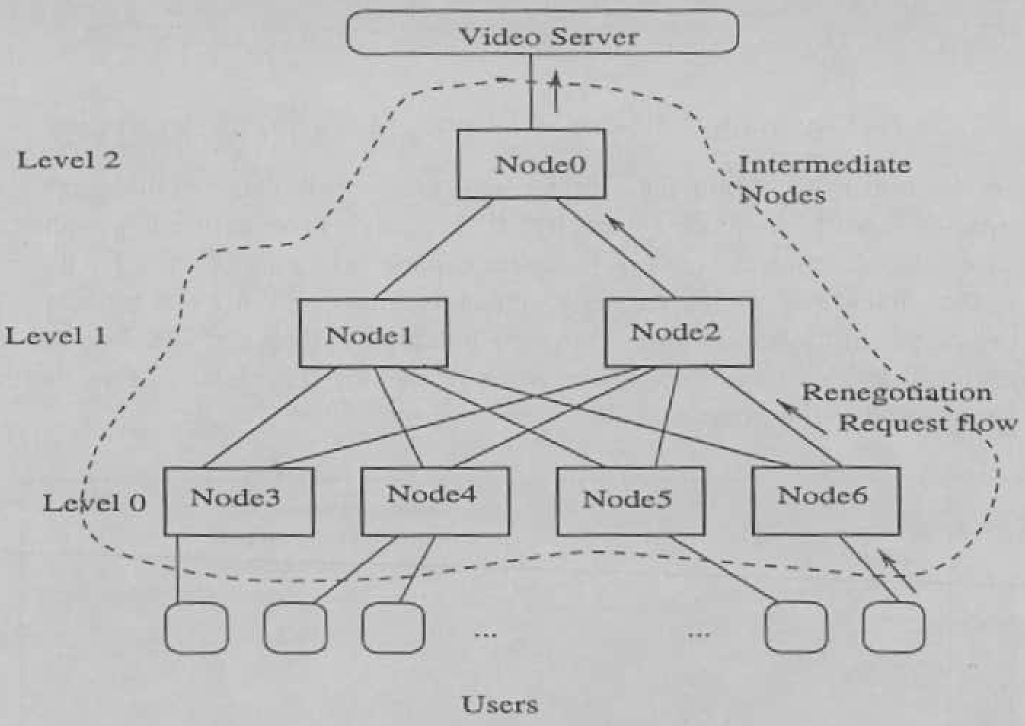

Figure 7: A Network with 7 nodes

These two types of networks are simulated using the basic link capacity (the parameter n described above) equal to 150 Mbps. The number of connections for each network is equal to 900. Figure 8 shows the comparative results for these two networks. For this part a renegotiation rejection occurs if any node in the path cannot support the new bandwidth level. As expected, in the balanced network all renegotiation rejections occur at nodes 3, 4, 5 and 6 and in the unbalanced networks all renegotiation rejections occur at node 0.

As can be seen in Figure 8 there are two distinct regions according to renegotiation rejection rates for balanced and unbalanced network. The bottleneck of the balanced network (nodes 3, 4, 5 and 6) is equivalent to one single node with eight output channels, each one with basic link capacity (150 or 300 Mbps). The bottleneck of the unbalanced network is the node 0 which has a single channel with capacity twice as great as the basic link capacity (150 or 300 Mbps). When the *maximum value* allowed for the *minimum bandwidth reservation* is greter than 4.5 Mbps, the renegotiation rejection rates on the unbalanced network is lower than those on the balanced network. The inverse situation occurs when the maximum value allowed for the minimum bandwidth reservation is lower than 4 Mbps. This result, that happens for both values of basic link capacity, shows that when the level allowed for minimum reservation is high the renegotiation rejection rate depend basically on the channel capacity. In other words, when the control applied by the minimum reservation scheme is strong the level of guarantee increases with the channel capacity. When such control is not applied (or when it is not so strong) the level of guarantee decreases with the channel capacity. Therefore, high bandwidth capacity is an important feature for networks supporting multimedia applications. But a mechanism for distributing this bandwidth in order to provide a high level of guarantee is also a fundamental feature for multimedia applications.

[1]the input and output sides are defined according to the renegotiation request flow only as topological references

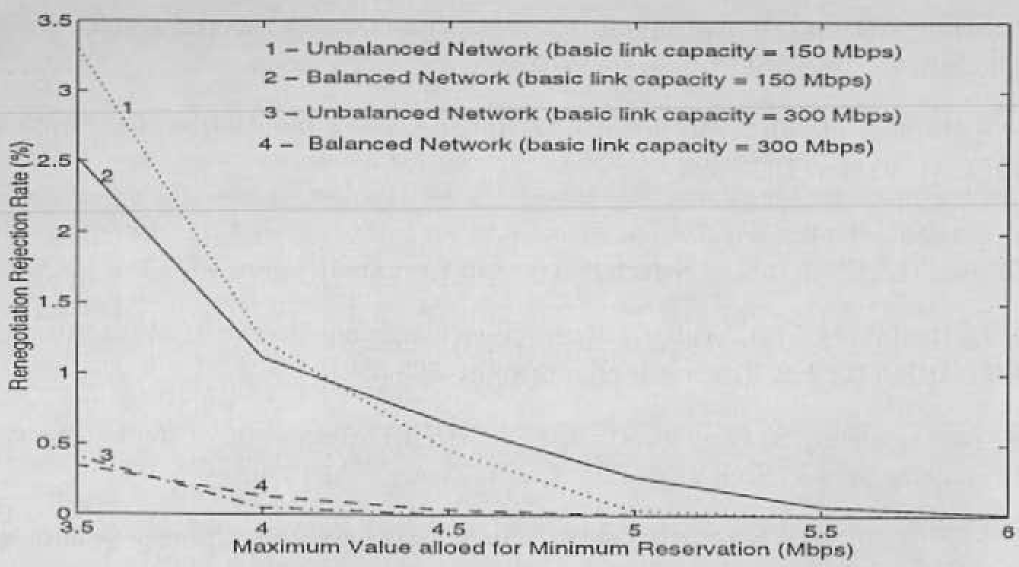

Figure 8: Comparison of Renegotiation Rates

5 Conclusions

This paper presented a general scheme for dynamic quality of service renegotiation at intermediate nodes. The proposed scheme explores variable bit rate characteristic of multimedia connections to improve resource utilization and consequently improving service availability. Some basic concepts related to multimedia applications and quality of service were revisited. The proposed scheme combines QoS adaptation and QoS renegotiation in order to create a new approach to perform QoS renegotiations.

An architecture for implementing the renegotiation scheme was presented and the basic algorithm for renegotiation was discussed. It was demonstrated that renegotiation scheme is feasible since the actual technology can support its processing needs. Besides that, the increase of traffic due renegotiation requests is under acceptable limits.

The proposed scheme is based on the specification of a minimum QoS level by each connection request in order to reserve resources that can be shared by all accepted connections. Minimum QoS is a statistical reservation scheme that results in a high guarantee level for all accepted connections since they will have available resources that can be used exclusively by future renegotiations.

A set of simulation programs was implemented and it has shown that renegotiation scheme can improve service rate when compared to peak allocation, keeping renegotiation rejections at a very low rate. Another important result is that this scheme depend on the link capcity. The greater the link capacity the lower is the renegotiation rejection rate. Since link capacities are increasing this scheme would increase its effectiveness.

The use of priority can improve even more this scheme with respect to guarantee levels and can better acommodate applications with different guarantee levels. Different classes of guaranteed and non-guaranteed services can co-exist under this scheme. The next step in our work will be to map the proposed scheme into ATM networks for implementation. The major goal of implementation is to consider some aspects related to real-time constraints that are dificult to observe in the simulation.

References

[ATM95] The ATM Forum Technical Committee, *ATM User-network Interface (UNI) Specification - Version 3.1*, Prentice Hall, 1995.

[Bla92] U. Black, *Network Management Standards:The OSI, SNMP and CMOL Protocols*, McGraw Hill, 1992.

[Cam93] A. Campbel, et. all., *Integrated Quality of Service for Multimedia Communications*, INFOCOM '93, pp. 732-739.

[Cou95] G. Coulson et all., *The Design of a QoS Controlled ATM-based Communication System in Chorus*, IEEE Journal of Selected Areas in Communication, vol 13, n. 4, May 1995.

[Dag94] A. F. Dagiuklas, M. Ghanbari, *Macromodeling Analysis of VBR Video in an ATM environment*, IEE 11th UK Teletraffic Symposium, 1994.

[Gro95] M. Grossglauser, S. Keshav, D. Tse, *RCBR: A Simple and Efficient Service for Multiple Time-scale Traffic*, SIGCOMM'95, Cambridge, USA, 1995.

[Fer92] D. Ferrari, J. Rmaekoers, G. Ventre, *Client Network Interaction in Quality of Service Communication Environment*,

[Haf95] A. Hafid, R. Dssouli - *A negotiation Model for Distributed Multimedia Application*, Publication Départamental n. 959, Départment IRO, Université de Montréal, March 1995.

[IETF96] *IETF Integrated Service documents (internet drafts)*, http://www.ietf.org/ids.by.wg/intserv.html, work in progress.

[Liu93] H. Liu, *Extension of Virtual Path Concept for QoS Guarantee*, Proceedings of Global Data Networking, pp. 114-119, 1993.

[Nah95] K. Nahrstedt, J. M. Smith, *The QOS Broker*, IEEE Multimedia, vol.2 n.1, dec. 1995.

[Ram92] J. Ramaekers, G. Ventre, *Quality of Service Negotiation in Real-Time Communication Networks*, TR-92-023, International Computer Science Institute, Berkely, April 1992.

[RFC1213] *RFC 1213 - Management Information Base for Network Management of TCP/IP-based Internets: MIB-II*, may 1990.

[Tou95] J. D. Touch, *Defining High-Speed Protocols: Five Challanges and an Example that Survives the Challenges*, IEEE Journal on Selected Areas in Communications, vol 13 n. 4, june 1995.

[Vog95] L. A. Vogel, B. Kerherve', G. von Bochmann, J. Gecsei, *Distributed Multimedia and QOS: A Survey*, IEEE Multimedia, Summer 1995.

[Zha93] L. Zhang, et al., *RSVP: A New Resource ReSerVation Protocol*, IEEE Network Magazine, vol7 n. 5, September 1993.