Integrated Resource Allocation for Real–Time Video Transfer to Maximize User's Utility

Kentarou FUKUDA, Naoki WAKAMIYA, Masayuki MURATA and Hideo MIYAHARA
Department of Informatics and Mathematical Science
Graduate School of Engineering Science, Osaka University
1–3 Machikaneyama, Toyonaka, Osaka 560–8531, Japan
E-mail: k-fukuda@ics.es.osaka–u.ac.jp

Abstract For providing distributed multimedia applications with end–to–end QoS (Quality of Service) guarantees, resource–reservation based control mechanisms should be employed in both of networks and end–systems. Resource reservation within the network can be achieved by virtue of bandwidth allocation mechanisms of Internet RSVP or ATM, and the CPU resource on the end system can be reserved by real–time OS. To achieve an effective use of resources while providing high quality video transfer, both resources of the network and server CPU should be allocated to clients in an integrated manner.

We first summarize the relationships among the video quality and the required amounts of CPU and network resources to provide a real–time video presentation. We find that high video quality can be kept by increasing network bandwidth even if CPU resources are not fully available. The opposite is also true. Based on these relationships, we next propose a new resource allocation scheme to share resources fairly among users by solving the utility maximization problem. In this paper, the utility is defined as functions of the user's benefit (video quality) obtained through allocated resources and cost paid for them. By solving the problem as an optimization problem, our scheme offers the adequate resource allocation based on their availabilities.

1 Introduction

With dramatic improvements in computing power, network bandwidth and video data compression techniques, there have been much advancements in distributed multimedia systems. The distributed multimedia system requires Quality of Service (QoS) guarantees in each entity within the system to perform effective and meaningful presentations [1, 2]. As a typical example, let us consider the distributed multimedia application, such as a teleconference or a live broadcast on the Internet, where video streams are coded by MPEG-2 (Moving Picture Experts Group) video coding algorithm [3]. In such an application, the MPEG-2 video is transferred from the video server to a number of clients through networks. The received video stream is de-compressed and presented on a computer display or a monitor. The video server should have a mechanism to encode and emit the requested video stream in a real–time fashion, and to communicate with clients interactively. The client must take care of the continuous and high–quality video presentation to users. The underlying transport network has to guarantee the network–level QoS; e.g., transfer delay, delay jitter and loss ratio.

For the network–level QoS, the resource–reservation based protocols such as a CBR (Constant Bit Rate) service class in ATM (Asynchronous Transfer Mode) [4] can provide the hard (or deterministic) guarantee. If the underlying network is the Internet RSVP [5, 6] can be employed. In our current study, the only assumption we make on the network is that the network has a capability to guarantee the bandwidth to the client.

In addition to the network–level QoS, the CPU–level QoS in terms of the processing delay and/or deadline violation ratio should be guaranteed in encoding and decoding MPEG-2 video. For this purpose, a sufficient amount of the CPU resource (i.e., processor cycles) should be reserved with an appropriate scheduling mechanism. However, traditional operating systems, which use the time–sharing scheduling algorithm like round–robin, cannot provide such mechanisms. As a result, the quality of real–time multimedia application cannot be guaranteed. Accordingly, a lot of researches have been recently devoted to real–time operating systems to provide CPU–level QoS guarantees [7-10]. Those real–time operating systems employ the real–time task scheduling algorithms which incorporate priority and/or deadline disciplines.

If both resource–reservation based networks and real–time operating systems are employed together, a real–time and high–quality multimedia distribution can be expected. To provide QoS guarantees in an effective way, however, a sufficient but not overbooked amount of network resources should be reserved and dedicated to the video distribution service. It is also true for reservation on server and client CPU resources. The problem on how to reserve the adequate amount of resources has been repeatedly pointed out and discussed as summarized below.

In resource–reservation based networks, such as ATM CBR service class, bandwidth allocation is performed at connection setup time. It implies that the required bandwidth must be known a priori or at least adequately estimated prior to the actual communication. In our previous study [2], we have proposed a QoS mapping method, which enables the multimedia system to predict the required bandwidth from QoS parameters in terms of spatial, SNR, and temporal resolutions. Once the bandwidth reservation succeeds, real–time video transfer can be achieved as far as the server and client have enough CPU resources. Of course, the sufficient CPU resources cannot be guaranteed in "best–effort" operating systems, and we need some real–time operating system, which makes it possible to reserve the CPU resources for the application. However, it implies that the amount of CPU resource to reserve must also be known or estimated at the service setup time. In [11], the authors propose predictors to estimate the required number of CPU cycles to decode MPEG com-

pressed video with software decoder. With those predictors, CPU cycles to decode the following frames or packets of the MPEG video stream can be estimated. However, it is difficult to dynamically reserve the CPU resource frame by frame or packet by packet. The video quality would be degraded if the reservation is rejected due to heavy CPU load. Furthermore, the authors do not take into account bandwidth needed to transfer the coded video stream. In addition to the above works, there have been many investigations on real–time operating systems [7-9] and real–time video transfer [12-14]. In those works, however, they consider the network resource and the CPU resource separately in spite of the fact that there must exist a strong relationship between them.

In our previous study [15], we have investigated relationships between the video quality and the CPU/network resources by the experimental analysis using actual MPEG-2 traces. From our analysis, we observed that if the network resource is sufficient and much bandwidth is offered to the connection, the server and clients can be free from complicated encoding/decoding tasks that require much server/client CPU resources. Conversely, the required bandwidth becomes small when the end systems can execute heavy coding tasks. These reasonable observations lead us to a flexible reservation mechanism based on availabilities of network and CPU resources. For this purpose we formulated the video quality as functions of network and CPU resources available to the user [15], which will be summarized in Section 2.

Results presented in Section 2 show that the video quality can be quantitatively related to network and CPU resources. The remaining problem is how those resources are allocated to users according to the availabilities of those resources. Since network and CPU resources are limited, those should be fairly allocated by some appropriate resource allocation scheme when either or both of resources are fully available to user's quality, which is our main subject of the current paper. In resource reservation based systems, the resource can be easily monopolized by some user when the user requests the unnecessarily large amount of resources. To prevent such a greedy resource acquisition, an appropriate control mechanism should be introduced. To consider fair allocation of resources, we introduce a user's utility defined by a function of user's benefit and cost for resources allocated to the application. In the case of video transfer, user's benefit can well be described by the perceived video quality achieved through the allocated resources as described above, and the cost can be regarded as the money or penalty paid for those resources. We then formulate the resource allocation mechanism to maximize users' utility as an optimization problem, which will be described in Section 3.

One point we should mention here is that in an actual situation, usable resources of clients are very diverse. CPU power and/or the bandwidth of access line to the Internet are very different among clients. Because of these heterogeneities of client environments, it becomes necessary to prepare a number of video streams to meet various requests of video qualities even for the single video source. An easiest way would be to provide many video streams according to each user's environment. However, it is obviously ineffective for the usage of server and network resources. In our previous work [16], we proposed a flow aggregation technique, which first gets together similar QoS requirements. Then, video streams are coded and transmitted for each group. By this mechanism, the required bandwidth and the number of video streams can be reduced. In [16], however, we only treated the minimization problem for the network bandwidth and did not consider heterogeneities of client environments. In this work, on the contrary, we first divide clients into several clusters according to the amount of available client CPU resource and bandwidth of access line by using K-means clustering [17], so that clients in the same cluster receive the same video stream transmitted on the multicast connection. In doing so, we will assume that the bottleneck within the network exists at the access line of the server, and then we allocate the bottleneck bandwidth and the server's CPU resource to each cluster while maximizing users' utility with consideration on the available resources.

The applicability of our scheme is then demonstrated by using the MPEG-2 video traces in Section 4. The results show that our scheme can provide users with high quality and effective real–time video transfer in the resource reservation based system. Our study can be applied to the existing real–time OS such as Tactix [10] or AQUA [8] which adjusts CPU resource allocation dynamically according to changes in availability of resources.

In this work, we assume a software codec to accomplish the truly dynamic and flexible coding control according to the optimal resource allocation. We would become free from considering CPU resources if we employ MPEG-2 hardware codec, but currently available products do not offer a dynamic and flexible parameter setting on resolutions and GoP structure. Without such functionality, we cannot expect an effective usage of network bandwidth. More important is that with hardware codec, the server cannot provide multiple video streams, which is suitable to heterogeneous environments.

This paper is organized as follows. In Section 2, we summarize the relationships between the video quality and the CPU/network resources required to provide a real–time video presentation. We then propose a new resource allocation scheme which maximizes users' utility by taking account of availability of resources in Section 3. In Section 4, we demonstrate the applicability of our scheme by using the MPEG-2 video traces. We conclude our paper in Section 5.

2 Relationship between Video Quality and Required Resources

In this section, we summarize relationships between the video quality and required CPU/network resources from our experimental analysis on actual MPEG-2 encoding/decoding traces [2, 15].

The required amount of resources and the quality of the MPEG-2 video are determined by coding parameters in terms

of spatial resolution R [pixels], SNR (Signal to Noise Ratio) resolution Q, temporal resolution, and GoP structure F [fps]. In our previous study [2], we proposed a QoS mapping method to predict the required bandwidth BW [Mbps] from those parameters. By using that method, the required bandwidth can be derived as:

$$BW(R,Q,F,G) \cong \left(\frac{1}{3.1}\right)^{\log_4 \frac{R}{640\times480}} \times \left(\alpha + \frac{\beta}{Q} - \frac{\gamma}{Q^2}\right) \frac{F}{30} BW_{base} \quad (1)$$

where G stands for the GoP structure. BW_{base} is a constant value corresponding to the required bandwidth of the reference sequence with parameter set $(R,Q,F,G) = (640 \times 480, 10, 30, G)$. BW_{base} decreases as the number of P/B pictures in GoP increases. Constant values of α, β and γ can be determined in the same way as in [2] according to the GoP structure G. For example, in the case of $G =$ 'I', we have those parameters as $\alpha = 0.151$, $\beta = 9.707$, and $\gamma = 4.314$. As shown in Eq. (1), the required bandwidth is proportional to temporal resolution F and a decreasing convex function of SNR resolution Q. When the number of pixels of each frame becomes four times larger, the required bandwidth BW is 3.1 times larger than that of the smaller video. Thus, once a set of encoding parameters is determined, we can estimate the required bandwidth.

Furthermore, we also investigate relationships among video parameters and required amount of CPU resource at end-systems [15]. From our investigations, the required CPU resource at the server side in terms of processor cycles, S, can be estimated by functions of spatial resolution R [pixels] and temporal resolution F [fps] as:

$$S \cong S_G \frac{R}{640\times480} \times \frac{F}{30} \quad (2)$$

where S_G is a constant value determined from the GoP structure G. S_G is proportional to the ratio of P/B pictures in the GoP structure. Because S_G differs among video sequences, an exact estimation of S_G is difficult. However the system is able to provide the QoS guarantee by using conservative estimation for S_G. Note that the SNR resolution does not affect the required amount of CPU resource at the server [2].

At the client side, the required number of CPU cycles C can be estimated from functions of the bandwidth BW [Mbps], spatial resolution R [pixels] and the GoP structure as:

$$C \cong BW \times 4.0\times10^7 + (8.7\times10^8 + \frac{N_p}{N}\delta + \frac{N_b}{N}\varepsilon) \times \frac{R}{640\times480} \times \frac{F}{30} \quad (3)$$

where N is the number of frames in the GoP structure, and N_p and N_b are numbers of P and B pictures in GoP, respectively. In Eq. (3), δ and ε are increasing rates against the amount of the CPU resource required to decode P and B pictures, respectively. We obtained $\delta = 2.8\times10^8$, $\varepsilon = 4.2\times10^8$ by applying the least–square approximation to the actual MPEG-2 traces [15].

From above observations, we can now see that the strong relationship exists among the video quality and the amounts of network/CPU resources. As an example, we depict Fig. 1 to see the effects of the SNR resolution and GoP structure on the required bandwidth. Each line in the figure is related to a specific quantization scale Q, and each point on the line corresponds to the GoP structure. When both the temporal and spatial resolutions are kept unchanged, the required bandwidth is largely determined by the GoP structure with same video quality in terms of SNR. Thus, the required amount of resources can be decreased by changing the GoP structure without degrading the video quality. For example, the required bandwidth decreases as the number of P/B pictures in GoP increases (see Fig. 1). However, as mentioned above, the required amounts of CPU resources at the end systems become larger as the number of P/B pictures in GoP increases. Those facts mean that there exists a clear trade-off between the network and CPU resources. Further, the larger quantization scale decreases the required amounts of two resources, bandwidth and CPU resource at the client. As shown in Fig 1, it is obvious that the required bandwidth is inversely proportional to the quantization scale. As a result, the required amount of CPU resource at the client C decreases as the required bandwidth BW decreases (see Eq. (3)).

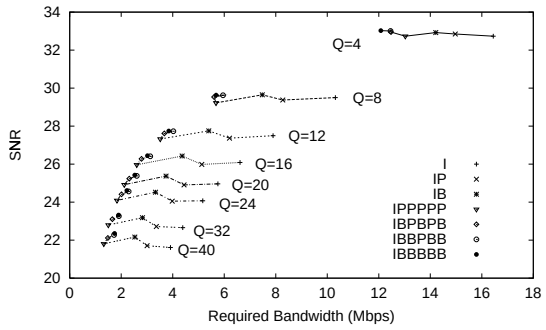

Figure 1: Relationship among GoP structure, quantization scale and video quality

From Eqs. (1) through (3), it is obvious that decreasing the temporal and/or spatial resolution can reduce the required amount of resources, but the video quality is considerably degraded [2]. By considering those relationships, we can determine a set of QoS parameters (including the GoP structure) which provides the high quality video transfer with limited resources. For instance, if the network resource is sufficient and the application can freely occupy much bandwidth, we do not have to use the GoP structure having P/B pictures. It means that the server and clients can be free from complicated encoding/decoding tasks of motion compensation which requires much CPU resource. Even if the network bandwidth becomes short, users can still enjoy high quality video presentation at the sacrifice of increased CPU cycles at the end-system. Based on these observations, we propose the utility–based resource allocation scheme by formulating this optimization problem in the next section.

3 Utility-based Resource Allocation Scheme

We have shown in the previous section that

1. video streams of identical QoS parameters (spatial, temporal and SNR resolution) have the same video quality regardless of GoP structures,
2. decreasing the spatial and/or the temporal resolution degrades the video quality while the required amount of resources decreases, and
3. increasing the quantization scale causes degradation of the video quality, while the required amounts of network bandwidth and the client's CPU resource decreases.

Keeping those facts in mind, we now introduce the user's utility function in the real–time MPEG-2 transfer.

3.1 Outline

Figure 2 depicts an example of the distributed multimedia system that our resource allocation scheme can be applied. Our scheme is performed as follows;

1. The QoS manager at the server manages an allocation of the network bandwidth and the server's CPU resource. It is performed in cooperation with (1) the bandwidth reservation mechanism provided by the resource–reservation based network and (2) the real-time OS at the server. At the connection setup time, each client notifies the server of how much CPU resource and bandwidth of access line are available for video tasks.

   The QoS manager then determines the amount of resources by considering the amounts of available resources of the server, the network and the clients. Its outline is summarized in the following three steps.

2. As mentioned before, A large number of video streams should be provided if the video stream is encoded according to the QoS request and the available resources of clients. Therefore, the QoS manager first divides clients into clusters by means of K-means clustering method [17] based on the available client resources. That is, clients of each cluster have similar characteristics on availabilities of resources, and each cluster is characterized by one representative parameter set of the bandwidth of the access line and the available CPU resource. Then, each cluster receives the same video stream using the multicast connection. Note that in our approach, each cluster is formed by the availability of resources, not by the geographical distances of clients.

   By this mechanism, both of the number of video streams that the server should provide and the total bandwidth for video streams can be much reduced. However, an effectiveness of clustering depends on the number of clusters and the system environment such as available resources and the number of clients. We have to provide an effective method to determine the number of clusters considering the system environment, and it remains future work.

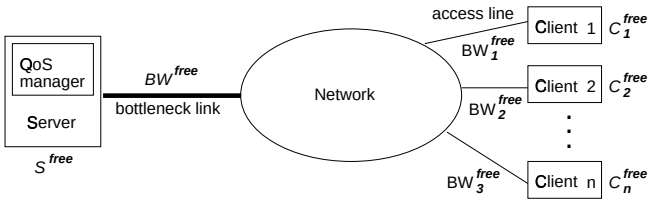

Figure 2: System model

3. The QoS manager then determines the amount of resources allocated to each cluster, so that total utility is maximized. (See 3.2 for a precise definition of our utility function.) In this paper, only the output link from the server is considered as the network resource to be allocated, since it is likely that the output link from the server becomes a bottleneck when the coded video streams are delivered to clients through the multicast links.

   Note that in utility maximization, the temporal and spatial resolutions are not changed, and the values specified by the application are just used because those have much influence on perceived video quality [2].

   In our method, the QoS manager first selects the candidate set of GoP structures such that total CPU resource consumption is within the available amount of the server's CPU resource. The resource allocation is then actually performed to maximize total utility with specified GoP assignment. In doing so, the value of SNR resolution is chosen appropriately according to our scheme. The utility maximization is examined for several GoP structures to find the optimal GoP structure. Then, the coding parameters of the video stream for each cluster are finally determined. More detailed descriptions will be given in Subsection 3.4.

4. The server starts to encode video streams in a real-time fashion with coding parameters determined according to our control scheme.

We note here that to measure the available network bandwidth, some signalling protocol is necessary. It is able to be implemented by, e.g., extending the RSVP protocol, but its precise mechanism is out of scope in this paper.

3.2 Definition of the Utility Function

We define the user's utility as a function of the user's benefit and the cost for resources allocated to the video transfer, where user's benefit can well be described by the perceived

video quality achieved through allocated resources as demonstrated in Section2, and the cost can be regarded as a penalty paid for those resources.

The utility of client i is defined as functions of the user's *Benefit* obtained through allocated resources and the *Cost* paid for them as follows;

$$U_i = Benefit_i(BW_i, S_i, C_i) - Cost_i(BW_i, S_i, C_i) \quad (4)$$

where BW_i means the bandwidth allocated to client i. S_i and C_i denote the amount of allocated CPU resources at the server and the client, respectively.

We next consider the cost function. Those clients that are allocated resources should pay the cost against those resources. Since it is necessary to inhibit users to become greedy, the cost is determined according to the user's share of available resources.

$$Cost_i(BW_i, S_i, C_i) = \frac{1}{N_{cluster_j}} \times \rho \frac{BW_i}{BW^{free}} + \frac{1}{N_{cluster_j}} \times \sigma \frac{S_i}{S^{free}} + \tau \frac{C_i}{C_i^{free}} + \upsilon \frac{BW_i}{BW_i^{free}} \quad (5)$$

where BW^{free}, S^{free} denote the available bandwidth and CPU resource at the server. C_i^{free} and BW_i^{free} denote the available CPU resource and bandwidth of access line at client i, respectively (see Fig. 2). $N_{cluster_j}$ is the number of clients in cluster j which client i belongs to. Since all clients belonging to the same cluster receive the same video stream, the bandwidth and CPU resource at the server side allocated to cluster j can be regarded as being devoted to all clients within the cluster. Thus, the cost for the bandwidth and CPU resource for cluster j are split among clients in the cluster j. As we described before, the client's CPU resource and bandwidth of access line are locally dedicated to the user. However, it is not adequate for the single application to occupy whole available local resources. Weighting factors ρ, σ, τ and υ take positive values to control the amount of consumed resources. Those should be appropriately determined by the QoS manager at the service setup time by taking account of the availability of resources. Of course, determination of these weighting factors is not an easy task. We will discuss this aspect in Subsection 4.2.

3.3 Derivation of Cluster's Utility

After clustering is performed, the QoS manager determines the resource allocation to maximize the clusters' utilities in total. The utility of cluster j can be described by the following equation;

$$\begin{aligned} U_{cluster_j} &= \sum_{i \in cluster_j} U_i \\ &= \sum_{i \in cluster_j} \{Benefit_i(BW_i, S_i, C_i) \\ &\qquad - Cost_i(BW_i, S_i, C_i)\} \end{aligned} \quad (6)$$

As we described in Subsection 3.2, clients in the same cluster receive the same video stream. Thus, the amount of shared resources that are used to encode and transmit the video stream to clients are same, and the local CPU resource that each client has to devote is identical. It follows that,

$$\begin{aligned} BW_{i \in cluster_j} &= BW_{cluster_j}, \\ S_{i \in cluster_j} &= S_{cluster_j}, \\ C_{i \in cluster_j} &= C_{cluster_j} \end{aligned} \quad (7)$$

By substituting Eq. (7) into Eq. (6),

$$\begin{aligned} U_{cluster_j} = & \\ \sum_{i \in cluster_j} & \{Benefit_i(BW_{cluster_j}, S_{cluster_j}, C_{cluster_j}) \\ & - Cost_i(BW_{cluster_j}, S_{cluster_j}, C_{cluster_j})\} \end{aligned}$$

where $BW_{cluster_j}$ is the bandwidth allocated to the cluster j, $S_{cluster_j}$ denotes the amount of the server's CPU resource allocated to cluster j, and $C_{cluster_j}$ stands for the amount of CPU resource required to decode the video stream at the client in cluster j. Since the benefit of users in the cluster j is also identical, we finally have

$$U_{cluster_j} = Benefit_{cluster_j} - Cost_{cluster_j} \quad (8)$$

$$\begin{aligned} Benefit_{cluster_j} &= N_{cluster_j} \\ &\quad \times Benefit(BW_{cluster_j}, S_{cluster_j}, C_{cluster_j}) \\ Cost_{cluster_j} &= \rho \frac{BW_{cluster_j}}{BW^{free}} + \sigma \frac{S_{cluster_j}}{S^{free}} \\ &\quad + \sum_{i \in cluster_j} \left\{ \tau \frac{C_{cluster_j}}{C_i^{free}} + \upsilon \frac{BW_{cluster_j}}{BW_i^{free}} \right\} \end{aligned}$$

By maximizing the sum of utilities of all clusters, clusters can fairly and effectively share the available resources.

3.4 Utility-based Resource Allocation Scheme

We now proceed to introduce our resource allocation scheme which maximizes the total utility under the restrictions on the network bandwidth and CPU resources at end systems. The problem can be formulated as:

$$\begin{aligned} \text{maximize} \quad & \textstyle\sum_j U_{cluster_j} \\ \text{subject to} \quad & \textstyle\sum_j BW_{cluster_j} \le BW^{free}, \\ & \textstyle\sum_j S_{cluster_j} \le S^{free}, \\ & C_{cluster_j} \le C_{cluster_j}^{free} \ \forall j, \\ & BW_{cluster_j} \le BW_{cluster_j}^{free} \ \forall j \end{aligned} \quad (9)$$

where $C_{cluster_j}^{free}$ and $BW_{cluster_j}^{free}$ denote the minimum amount of available CPU resource and the bandwidth of access line at clients in the cluster j. Thus, those are expressed as follows:

$$\begin{aligned} C_{cluster_j}^{free} &= \min\{C_i^{free} | i \in cluster_j\}, \\ BW_{cluster_j}^{free} &= \min\{BW_i^{free} | i \in cluster_j\} \end{aligned} \quad (10)$$

To solve the maximization problem of Eq. (9), we should have the knowledge about the characteristics of $Benefit_{cluster_j}$

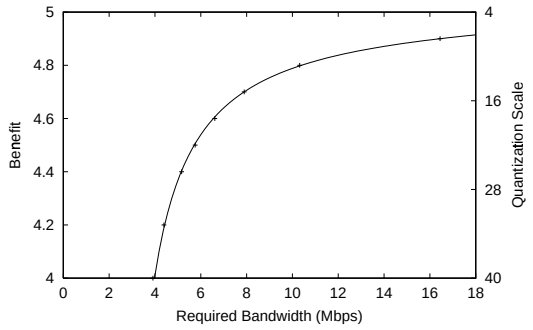

Figure 3: Relationship among quantization scale, required bandwidth and user's benefit

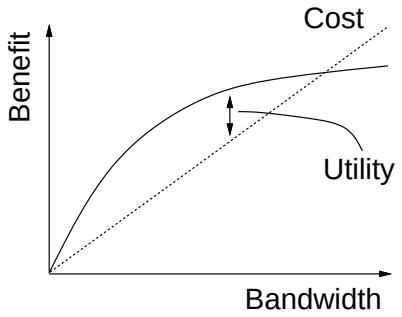

Figure 4: Relationship between allocated bandwidth and benefit with cost function

and $Cost_{cluster_j}$. In our work, the user's benefit is described in terms of MOS (Mean Opinion Score) since MOS values directly reflect the perceived video quality. As having been discussed in Section 2, the required resources (i.e., BW_i, S_i and C_i) can be represented by the coding parameters R_i, Q_i, F_i, and G_i of the MPEG-2 video (see Eqs. (1) through (3)). Therefore, the benefit function $Benefit_i(BW_i, S_i, C_i)$ of the above equation can be rewritten as $Benefit_i(R_i, Q_i, F_i, G_i)$.

In our previous work [2], we have observed that the MOS value is a monotonically increasing function of each QoS parameter, but is not necessarily continuous and concave. This fact implies that the optimal solution cannot be obtained by a direct mathematical operation. We therefore introduce some assumptions on the set of QoS parameter. Considering the fact that the temporal and spatial resolutions have much influence on the perceived video quality [2], we assume that those parameters are deterministically specified by the QoS manager, considering the characteristics of application and video contents. Once those QoS parameters are specified, the amount of server's CPU resource becomes a function of the GoP structure alone (see Eq. (2)). In addition, the required amount of the CPU resource at the client can be derived from Eq. (3) as a function of the allocated bandwidth and the GoP structure. Thus, in the case where the GoP structure $G_{cluster_j}$ is used for cluster j, we now rewrite the benefit function only as a function of the bandwidth, i.e.,

$$Benefit_i(BW_i, S_i, C_i) = Benefit_{G_{cluster_j}}(BW_{cluster_j}) \quad (11)$$

Substituting Eq. (11) to the benefit function of cluster in Eq. (8) yields

$$Benefit_{cluster_j} = N_{cluster_j} \times Benefit_{G_{cluster_j}}(BW_{cluster_j}) \quad (12)$$

Hereafter, we investigate the detailed relationship between the cluster's benefit $Benefit_{cluster_j}$ and the required bandwidth $BW_{cluster_j}$. Figure 3 depicts the relationship among the SNR resolution (i.e., the quantization scale) and the user's benefit in terms of MOS values and required bandwidth. Points in the figure are obtained through the actual MOS evaluation involving seven testees. In the evaluation, the GoP structure consists of only I pictures. From Fig. 3, we can observe that the user's benefit is a monotonically increasing concave function of allocated bandwidth. While not shown in the figure, the same tendencies were observed in other GoP structures and resolutions. The basic properties are summarized as follows.

Property 1 *The benefit function $Benefit_{G_{cluster_j}}$ is continuous and differentiable with respect to the bandwidth.*

Property 2 *User's benefit is a monotonically increasing concave function of allocated bandwidth (see Fig. 3).*

$$\frac{\partial Benefit_{G_{cluster_j}}(BW_{cluster_j})}{\partial BW_{cluster_j}} > 0,$$
$$\frac{\partial^2 Benefit_{G_{cluster_j}}(BW_{cluster_j})}{\partial BW^2_{cluster_j}} < 0$$

When the allocated bandwidth increases, the perceived video quality gets higher. However, the increase of the bandwidth becomes less effective as the allocated bandwidth is large.

Similarly to the case of the benefit function, the cost function is rewritten as:

$$Cost_{cluster_j} = \rho \frac{BW_{cluster_j}}{BW^{free}} + \sigma \frac{S_{G_{cluster_j}}}{S^{free}} + \sum_{i \in cluster_j} \left\{ \tau \frac{C_{G_{cluster_j}}(BW_{cluster_j})}{C_i^{free}} + \upsilon \frac{BW_{cluster_j}}{BW_i^{free}} \right\} \quad (13)$$

where $S_{G_{cluster_j}}$ is a constant value only dependent on the GoP structure. $C_{G_{cluster_j}}$ denotes a function of the allocated bandwidth and is a monotonically increasing linear function (Eq. (3)). From Eq. (13), we can obtain the following equations by differentiating the cost function with respect to the allocated bandwidth $BW_{cluster_j}$.

$$\frac{\partial Cost_{cluster_j}}{\partial BW_{cluster_j}} = \theta \geq 0, \quad \frac{\partial^2 Cost_{cluster_j}(BW_{cluster_j})}{\partial BW^2_{cluster_j}} = 0 \quad (14)$$

where θ denotes a gradient of the cost function.

By using Eqs. (12) and (13), the cluster's utility is now defined as a function of the allocated bandwidth for given GoP structure $G_{cluster_j}$. From Properties 1 and 2, the relationship between the allocated bandwidth and user's utility can be described by using Eqs. (14) as;

$$\frac{\partial U_{G_i}}{\partial BW_i} = \frac{\partial Benefit_{G_i}(BW_i)}{\partial BW_i} - \theta,$$
$$\frac{\partial^2 U_{G_i}}{\partial BW_i^2} = \frac{\partial^2 Benefit_{G_i}(BW_i)}{\partial BW_i^2} < 0 \quad (15)$$

These relationships are also shown in Fig. 4. As shown in Fig. 4, the utility is substantially increased when the allocated bandwidth is relatively small. On the other hand, the cost is increased faster than the video quality when the sufficient bandwidth has already been allocated. Those results come from the relationship between the "linear" cost function and the "concave" benefit function.

Because $U_{cluster_j}$ is a statically concave function, the maximization problem for the total utilities of all clusters (Eq. (9)) can be re–stated by the following optimization problem.

$$\text{maximize} \sum_i U_{cluster_j} = Benefit_{cluster_j} - Cost_{cluster_j} \quad (16)$$

$$\begin{aligned} \text{subject to} \quad & \textstyle\sum_j BW_{cluster_j} \le BW^{free}, \\ & \textstyle\sum_j S_{G_{cluster_j}} \le S^{free}, \\ & C_{G_{cluster_j}}(BW_{cluster_j}) \le C^{free}_{cluster_j} \quad \forall j, \\ & BW_{cluster_j} \le BW^{free}_{cluster_j} \quad \forall j \end{aligned}$$

As mentioned before, the required CPU resource at the server, $S_{G_{cluster_j}}$, is constant for given GoP structure $G_{cluster_j}$. Thus, we can assume that the QoS manager computes the total CPU resource required at the server a priori. When the resultant $\sum_j S_{G_{cluster_j}}$ exceeds the available CPU resource S^{free}, the QoS manager needs to force the cluster (i.e., the user whose $S_{G_{cluster_j}}$ is largest) to change the GoP structure to decrease $\sum_j S_{G_{cluster_j}}$. By using it as the first step of an optimization procedure, we do not have to consider the restriction on the CPU resource at the server again.

The constraint on the total bandwidth allocated is represented as $\sum_j BW_{cluster_j} \le BW^{free}$. It can be applied by using a barrier function such as

$$Barrier_{BW} = \frac{1}{BW^{free} - \sum_j BW_{cluster_j}} \quad (17)$$

As far as the total allocated bandwidth $\sum_j BW_{cluster_j}$ does not exceed the available bandwidth BW^{free}, $Barrier_{BW}$ is a monotonically increasing and strictly convex function of the total allocated bandwidth. In the same way, the constraint on the client's CPU resource and bandwidth of access line allocated to users can be written as:

$$\begin{aligned} Barrier_C = & \quad \textstyle\sum_j \frac{1}{C^{free}_{cluster_j} - C_{G_{cluster_j}}(BW_{cluster_j})}, \\ Barrier_{BW_{access}} = & \quad \textstyle\sum_j \frac{1}{BW^{free}_{cluster_j} - BW_{cluster_j}} \end{aligned} \quad (18)$$

Then, the maximization problem can be formulated by the following equation;

$$\begin{aligned} \max\{ & \sum_j U_{cluster_j} - Barrier_{BW} \\ & - Barrier_C - Barrier_{BW_{access}} \} \end{aligned} \quad (19)$$

From Eq. (15), each cluster's utility is a strictly concave function of the bandwidth ($BW_{cluster_j}$) allocated to the users in the cluster. Further, $Barrier_{BW}$, $Barrier_C$ and $Barrier_{BW_{access}}$ are monotonically increasing and strictly convex functions of $BW_{cluster_j}$. Then, our problem has an unique and optimal solution. The optimal condition can be described by the following equation.

$$\begin{aligned} & \frac{\partial \sum_j U_{cluster_j}}{\partial BW_{cluster_j}} - \frac{\partial Barrier_{BW}}{\partial BW_{cluster_j}} - \frac{\partial Barrier_C}{\partial BW_{cluster_j}} \\ & \qquad - \frac{\partial Barrier_{BW_{access}}}{\partial BW_{cluster_j}} = 0, \quad \forall i \end{aligned} \quad (20)$$

When the bandwidth allocation satisfies the above equation, maximization of both the user's utility and the total utility are assured. We can solve this problem by using a descent method such as Newton's method.

4 Numerical Examples and Discussions

4.1 Applicability of Proposed Resource Allocation Scheme

In this subsection, we demonstrate the applicability of our scheme using MPEG-2 video traces. We use the network model depicted in Fig. 2 where the output link of the server is the bottleneck link. The QoS manager exists in the server in order to manage the resources on the bottleneck link BW^{free} and the server's CPU S^{free}. The number of clients is set to be 200. Each client knows its own available CPU resource C_i^{free} and bandwidth of access line BW_i^{free}. The client's benefit is given as a function of MOS (Fig. 3). For sake of simplicity, we assume that clients request the server of the single video contents.

In evaluation, the available resources on the bottleneck link and the server's CPU are set to be $BW^{free} = 50$ [Mbps] and $S^{free} = 20$ [Gcycles/sec], respectively. The available amount of CPU resource at the client C_i^{free} is chosen at random from 1.1 to 2.0 [Gcycles/sec]. The bandwidth of access line BW_i^{free} is also randomly chosen from a range of 3 to 20 [Mbps]. The number of cluster is set to be 6. The spatial and temporal resolution of video are fixed at 640x480 pixels and 30 fps, respectively, for ease of presentation. The characteristics of resources at clusters are summarized in Table 1.

In performing our allocation scheme, the quantization scale is varied from 4 (highest SNR) to 40 (lowest). The candidate GoP structures are 'I', 'IP', 'IB', 'IPPPPP', 'IBPBPB', 'IBBPBB' and 'IBBBBB'. The weighting parameters ρ, σ, τ and υ are assumed to be identical and fixed at 0.2. Note that more discussions on weighting parameters will be presented in Subsection 4.2.

The result is summarized in Table 2, where we show the assigned GoP structure, the cluster's utility, the client's benefit and the utilization ratio of resources (determined by our allocation scheme). From Tables 1 and 2, we can observe that our scheme can offer adequate resource allocation based on the availabilities of resources. For instance, clusters 1 and 2, which do not have sufficient CPU resources (about 1.2 [Gcycle/sec]) at the client, can enjoy high quality video presentation by using a relatively large amount of resources. Clients of cluster 2 do not have sufficient access line bandwidth, and therefore our scheme allocates the large server

Table 1: Characteristics of clusters

cluster	# of clients	$C^{free}_{cluster_j}$ [Gcycle/sec]	$BW^{free}_{cluster_j}$ [Mbps]
1	32	1.17	11.50
2	33	1.18	3.05
3	31	1.37	7.62
4	26	1.41	3.33
5	53	1.48	12.58
6	25	1.75	3.44

Table 2: Utilization ratio of resources, selected GoP structure and utility of clusters

cluster	selected GoP structure	cluster's utility	benefit	utilization ratio (%)			
				bottleneck bandwidth	server CPU	client CPU	access line bandwidth
1	I	15.30	0.76	15.33	3.35	90.30	50.16
2	IPPPPP	11.22	0.60	4.72	26.50	92.65	32.23
3	IBPBPB	16.68	0.85	13.14	11.50	89.75	64.29
4	IBPBPB	11.40	0.74	6.64	20.50	82.47	62.35
5	IB	35.92	0.95	15.68	20.50	85.08	48.11
6	IBBBBB	12.00	0.73	6.88	17.45	74.55	42.75

Table 3: Total of users' utility, average benefit and utilization ratio of resources

proposed scheme	total utility	average benefit	utilization ratio (%)			
			bottleneck bandwidth	server CPU	client CPU	bottleneck bandwidth
with	102.510	0.792	62.382	99.800	86.231	49.504
without	64.241	0.522	4.134	26.500	75.148	18.370

CPU resource in order to reduce the required resources at the client side. On the other hand, clients of cluster 1 can enjoy higher–quality video presentation owing to the large bandwidth. It is because in this case, the end-systems can be free from complicated coding tasks for encoding/decoding of P/B pictures. In our previous study [15], we observed that the required bandwidth decreases as the number of P/B pictures in GoP increases. However, the required amounts of CPU resources at the end-systems become larger as the number of P/B pictures in GoP increases as mentioned in Section 2. In Tables 1 and 2, we can see that our scheme can offer the adequate resource allocation based on the compromise between the network and CPU resources.

In [15], we also observed that P pictures require a larger amount of server's CPU resource than B pictures, but the required amount of client's CPU resource is less than that of B pictures. Such a trade-off relationship between the CPU resources at the server and clients can actually be taken into account in our scheme. Clusters 4 and 6 have almost the same available access line bandwidth (about 3.4 [Mbps]), but the available amounts of CPU resources at clients are different. Our resource allocation scheme then selects the GoP structure "IBPBPB" for cluster 4. On the other hand, the GoP structure "IBBBBB" requiring less server CPU resource and more client CPU resource is chosen for cluster 6. That is, when the available amounts of client CPU resources become insufficient, the GoP structure with less B pictures is chosen to decrease the required amounts of client CPU resources without the degradation of clients' benefit. As a result, total cost for the video transfer can be decreased though the required server CPU resource is increased. This result comes from the characteristics of our cost function (Eq. (5)). The cost for the insufficient resource is increased faster than that of other resources. On the other hand, as the available amounts of the client CPU resource become large, the GoP structure with more B pictures is chosen in order to decrease the required amounts of the server's CPU resource. The remaining server's CPU resource are distributed to other clusters, whose resources are more insufficient, to improve clients' benefit (i.e., video quality) and reduce the required amount of insufficient resources. These results reflect the fact that the client's utility is derived as the subtraction of the cost from the benefit.

Table 3 shows the total of clients' utility, average of clients' benefit and utilization ratio of resources. For comparison purposes, we also consider the case where the system provides all clients with the single video stream which maximizes the benefit with the minimum amount of available resources among all clients, i.e., 1.17 [Gcycle/sec] for client CPU and 3.05 [Mbps] for access line bandwidth(see Table 1). This is, in a sense, a coward strategy to avoid over-consumption of shared resources, which easily occurs with a greedy strategy. The single video stream with the GoP structure of "IPPPPP" is distributed to all clients. From Table 3, we can observe that clients can enjoy high quality (in terms of benefit) video presentation due to the effective resource allocation

with our scheme. On the other hand, without our scheme, the system cannot provide clients with high quality video presentation. It is because that shared resources are not effectively utilized. As a result, total of clients' utility also becomes less than that obtained by our scheme.

From results presented above, we can conclude that our scheme can offer the reasonable resource allocation based on the resource availabilities and the trade-off among the network and CPU resources.

4.2 On Weighting Factors

As stated in Subsection 3.2, the weighting factors ρ, σ, τ and υ in Eq. (5) are the positive values to control the amount of consumed resources. If one of the weighting factor is set to 0, the corresponding resource is regarded as free, and can be consumed more to achieve the larger benefit. As the weighting factor becomes larger, the QoS manager tries to maximize the total utility with a smaller amount of the expensive resource. As a result, the utilization of the resource with the increased weight becomes smaller than that with the smaller weight.

For example, the QoS manager can hold down the usage of bottleneck bandwidth by setting a larger weighting factor ρ. It implies that the system environments should be taken into account in determining the weighting factors. If the multimedia system is distributed over the public network and the bandwidth is expensive resource, the factor ρ and υ should be large enough to avoid using expensive bandwidth. On the other hand, in the local area environment, the bandwidth is free. Thus, the parameters ρ and υ should be small to allocate the bandwidth as much as possible. As a result, the video quality can be improved as high as possible within the available amount of CPU resources.

5 Concluding remarks

In this paper, we first summarize relationships among the video quality and the amount of network/CPU resources in providing a real–time video communication. Based on these relationships, we propose a new resource allocation scheme which fairly allocates the network bandwidth and the server's CPU resource while maximizing users' utility with consideration on the available resources. Through numerical examples, we have shown the applicability of our scheme by using the MPEG-2 video traces.

In this work, we first divide clients into several clusters according to the amount of available client CPU resource and bandwidth of access line by using K-means clustering. However, the effectiveness of clustering depends on the number of clusters and the system environment, such as available resources and the number of clients. We need further work on investigating the relationships among them and determination of the number of clusters. Further, in this paper, we select the set of GoP structures by an exhaustive search. It requires much computing power to determine the optimal allocation of resources. We believe that a heuristic search can derive a suboptimal allocation within much less computing power than an exhaustive search. However, investigation of the effective algorithm for a heuristic search remains future research topic.

Now, we are working on implementation of video transfer system with our resource allocation scheme. System consists of PCs (Pentium III 500Mhz) running real–time OS "Tactix" [10] as endsystems and 100base-T and/or Gigabit ethernet for the network where RSVP is employed. Unfortunately, our MPEG-2 software encoder currently can't perform real–time coding. Hence, we first prepare coded video streams on the server's storage and the server's CPU resource is virtually assigned. The clients currently can decode 320x240 pixels large video stream at about 24 fps in real–time.

Acknowledgments

This work was partly supported by Special Coordination Funds for promoting Science and Technology of the Science and Technology Agency of the Japanese Government, Research for the Future Program of Japan Society for the Promotion of Science under the Project "Integrated Network Architecture for Advanced Multimedia Application Systems," Telecommunication Advancement Organization of Japan under the Project "Global Experimental Networks for Information Society Project," and a Grant-in-Aid for Encouragement of Young Scientists 10750277 from The Ministry of Education, Science, Sports and Culture of Japan.

References

[1] A. T. Campbell, G. Coulson, and D. Hutchison, "A quality of service architecture," *ACM Computer Communication Review*, vol. 24, pp. 6–27, April 1994.

[2] K. Fukuda, N. Wakamiya, M. Murata, and H. Miyahara, "QoS mapping between user's preference and bandwidth control for video transport," *Proceedings of Fifth IFIP International Workshop on Quality of Service '97*, pp. 291–302, May 1997.

[3] ISO/IEC DIS 13818-2, "MPEG-2 video," *ISO standard*, 1994.

[4] ITU-T Recommendation I.371, "Traffic control and congestion control in B-ISDN," *International Telecommunication Union*, 1992 revised in 1995.

[5] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A new Resource reSerVation Protocol," *IEEE Network Magazine*, vol. 7, pp. 8–18, September 1993.

[6] P. P. White, "RSVP and integrated services in the Internet: A tutorial," *IEEE Communications Magazine*, pp. 100–106, May 1997.

[7] C. W. Mercer, S. Savage, and H. Tokuda, "Processor capacity reserves: Operating system support for multimedia applications," *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, pp. 90–99, May 1994.

[8] K.Lakshman and R. Yavatkar, "Integrated CPU and network-I/O QoS management in an endsystem," *Proceedings of Fifth IFIP International Workshop on Quality of Service*, pp. 167–178, May 1997.

[9] C. Lee, R. Rajkumar, and C. Mercer, "Experiences with processor reservation and dynamic QOS in Real-Time Mach," *Proceedings of Multimedia Japan*, March 1996.

[10] M. Iwasaki, T. Takeuchi, M. Nakahara, and T. Nakano, "Isochronous scheduling and its application to traffic control," *Proceedings of 19th IEEE Real-Time Systems Symposium '98*, pp. 14–25, December 1998.

[11] A. Bavier, B. Montz, and L. Peterson, "Predicting MPEG execution times," *Proceedings of ACM SIGMETRICS '98*, pp. 131–140, June 1998.

[12] S. Singh and S. Chan, "A multi-level approach to the transport of MPEG-coded video over ATM and some experiments," *Proceedings of IEEE GLOBECOM'95*, pp. 1920–1924, November 1995.

[13] D. P. Heyman and T. V. Lakshman, "What are the implications of long-range dependence for VBR-video traffic engineering?," *IEEE/ACM Transactions on Networking*, vol. 4, pp. 301–317, June 1996.

[14] J. C. Wu, Y. Chen, and K. Jiang, "Modeling and performance study of MPEG video sources over ATM networks," *Proceeding of IEEE ICC'95*, pp. 1747–1750, June 1995.

[15] K. Fukuda, N. Wakamiya, M. Murata, and H. Miyahara, "QoS guarantees based on end-to-end resource reservation for real-time video communications," *Proceedings of 16th International Teletraffic Congress*, pp. 857–866, June 1999.

[16] K. Fukuda, N. Wakamiya, M. Murata, and H. Miyahara, "On flow aggregation for multicast video transport," *Proceedings of Sixth IFIP International Workshop on Quality of Service '98*, pp. 13–22, May 1998.

[17] J. A. Hartigan, *Clustering algorithms*. John Wiley & Sons, 1975.