Integrated Resource Allocation Scheme for Real-Time Video Multicast

Naoki Wakamiya Taketo Yamashita Masayuki Murata Hideo Miyahara
Department of Information Networking,
Graduate School of Information Science and Technology, Osaka University
1–3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan
Tel: +81–6–6850–6586 Fax: +81–6–6850–6589
E-mail: wakamiya@ist.osaka-u.ac.jp

***Abstract*— To provide distributed multimedia applications with end-to-end QoS (Quality of Service) guarantees, resource reservation-based control mechanisms should be employed in both networks and end systems. In this paper, we propose a resource allocation scheme for real-time video multicasting described as a utility maximization problem. In this scheme, clients are first divided into multicast groups by means of a clustering technique. Then system resources are allocated to each group so that the total utility is maximized. We have confirmed that our proposed scheme can achieve effective use of resources while providing high-quality video to users.**

I. Introduction

Due to dramatic improvements in computing power, network bandwidth, and video data compression techniques, distributed and real-time multimedia systems are now widely used. In these systems, a server captures video data, then encodes and sends it to clients via networks. The clients receive the coded data, decode it, and display it to users. To provide the users with a high-quality multimedia presentation, QoS should be guaranteed in terms of the data transfer delay, and the regularity of the video encoding and decoding [1]. Network level QoS, such as packet loss ratio and transfer delay, can be statically guaranteed in bandwidth reservation-based networks [2]. A real-time OS that reserves and schedules CPU resources can provide high-speed, high-quality video coding and decoding on end systems [3, 4].

However, even if we can successfully build a distributed multimedia system by combining these platforms, high-quality, real-time video transfer cannot be achieved efficiently without appropriate prediction and reservation mechanisms for both the network and end systems resources. We have formulated the effect of the MPEG-2 coding parameters on the required resources and the video quality, and we found that there is a strong relationship between them [5]. Based on these relationships, the resource allocation scheme proposed in [6] enables high-quality video transfer within limited resources by maximizing a user's "utility", which is represented as a relationship between the "benefit" obtained through allocated resources and the "cost" paid for them. We verified the practicality of the scheme by implementing it on an actual video distribution system. However, we also found that it cannot be applied to more general networks where heterogeneity also exists in the available bandwidth, because only the server's access link is taken into account and limitations on the other links including the clients' access links are assumed to be negligible.

In this paper, based on our previous work, we propose a new resource allocation scheme for video multicast systems with heterogeneous clients. We take into account a network topology, including the locations of the server and clients and the link bandwidth. Our scheme first divides the clients into multicast groups, then the shared resources are allocated to each group in an integrated manner based on the relationships between the resources. By iteratively repeating "local and global resource allocations", the total utility is maximized. Appropriate multicast groups are thus established, so that users are provided with a video stream of the highest possible quality. We evaluate applicability and efficiency of the proposed scheme by applying it to a multicast network models with clients that are heterogeneous with regard to their available amounts of resources.

This paper is organized as follows. In Section II, we describe the multicast system we consider, and we describe our clustering and multicast tree construction methods in Section III. Then we briefly introduce the relationships between video quality and the required resources, and outline our resource allocation scheme in Section IV. In Section V, we evaluate the effectiveness of the proposed scheme. We finally conclude our paper in Section VI.

II. Scenario of Video Multicast System

The network we assume has a general topology and is capable of multicasting. The amounts of available bandwidth are diverse and differ among links. We assume that the network offers bandwidth reservation mechanisms such as ATM, DiffServ, IntServ or TTCP/ITM [4]. The CPU resources can be controlled and reserved with a real-time OS such as Real-Time Mach [7] or HiTactix [4]. We assume that heterogeneous end systems are involved in the video multicast, and that the available amounts of CPU resources also vary.

In our system, the server first notifies users about the video session, including the starting time, the subscription due time, and the contents of the service, either through broadcasting or by using a dedicated multicast address as in SDP [8]. Then the client for a user intending to join reserves the available CPU resources and informs the server of the amount successfully reserved. At the same time, the server examines its own resource availability and the network conditions (i.e., the available bandwidth of the links), through a bandwidth reservation protocol. Based on this information, the server then composes multicast

groups by dividing the clients into clusters based of their available access link bandwidth and CPU resources. The server then constructs multicast trees.

Next, the shared resources—i.e., the server CPU and the link bandwidth—are allocated to each session. Initially, identical amounts of shared resources are allocated to each multicast group sharing the same bottleneck link. In each group, resource allocation is performed so as to maximize the quality of the video transfer, given the relationship between resources (local resource allocation). Then, the remaining resources are re-allocated to the cluster, which is expected to contribute to increasing the total utility (global resource allocation). This two-level resource allocation is formulated as a utility maximization problem in which the utility is represented as a relation between the benefit and the cost, i.e., the obtained video quality and the required amount of resources.

By iteratively repeating these global and local utility maximizations, the server determines the resource allocation. Then, it reserves its own CPU resources, reserves bandwidth for each session by a method such as RSVP, and notifies the clients of the required amounts of CPU resources. Each client confirms it has reserved the required CPU resources, and real-time video multicasting begins.

III. Multicast Session Construction by Clustering

There are several approaches to tackling the resource heterogeneity, such as a simulcast a layered multicast [9], and active networks [10]. Our approach is similar to a simulcast in that the server generates independent video streams of varying quality. The number of multicast groups is reduced through a clustering technique, a video stream of appropriate quality is chosen for each cluster, and resources are adequately allocated among clusters and system entities.

Each cluster corresponds to a multicast group carrying a video stream of a quality appropriate for the group's members. Although heterogeneity exists in the available bandwidth in the network for video transfer, in the clustering phase we only take into account the CPU resources and access link bandwidth. This is because the available bandwidth for the multicast session cannot be determined until the multicast trees are constructed.

A. Clustering of clients

There are several clustering algorithms for grouping samples by their similarities [11]. The k-mean algorithm is one widely known clustering algorithms. It generates k clusters based on Euclidean distance and is favored for its simplicity. Since the initial k points are chosen at random in the k-mean clustering algorithm, the speed of convergence and the feasibility of the obtained clusters vary from trial to trial [12]. The KA algorithm was thus proposed to avoid the instability of the k-mean clustering algorithm, and it have been verified that the KA algorithm obtains a unique initial state that leads to more centralized clusters [12, 13].

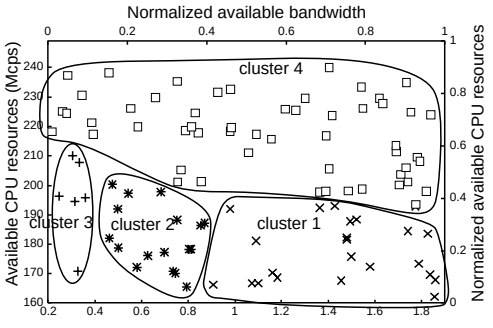

Fig. 1. Linear normalization

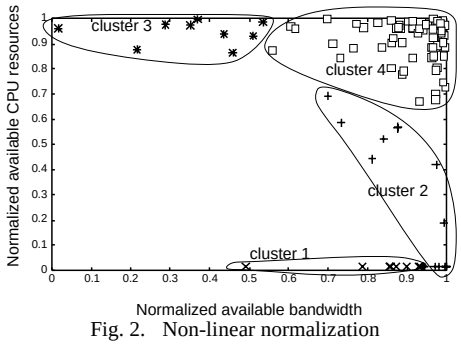

Fig. 2. Non-linear normalization

To apply the KA and k-mean algorithms, we must first determine the number of clusters, k. We repeatedly try clustering over a range of values from $k = 1$—that is, all clients are accommodated in a single multicast group and provided with a single video stream—to a specified maximum number of clusters, say, K. In an extreme case, K is identical to the number of clients and causes a scalability problem. In an actual situation, however, K is limited to a realistic number since the system resources cannot accommodate too many simultaneous multicast sessions.

B. Mapping client resources

To apply a clustering algorithm to grouping heterogeneous clients on the basis of their resource availabilities, the amounts of available resources must be normalized to a range from 0 to 1. We take the access link bandwidth and client CPU resources into account in clustering the clients. Since the possible combinations of these two parameters, which are determined from a set of coding parameters such as the quantizer scale and the GoP structure, does not form a linear function, we apply a non-linear normalization derived from the relationships between the video quality and the required resources. An example of a comparison between the linear and non-linear normalization methods is shown in Figs. 1 and 2. Fig. 1 depicts the distribution of the clients in accordance with their available resources and the resultant clusters with a linear normalization. With non-linear normalization, the resulting clusters change as shown in Fig. 2.

C. Multicast tree construction

For a high-quality, efficient video multicast, the multicast tree of each cluster is constructed to contain less number of

links that have larger available bandwidth. The Steiner Tree problem consists of searching the k-MST (Minimum Spanning Tree) spanning to specific nodes k, and this problem is NP-complete [14]. We define the cost of a link as the reciprocal of the available bandwidth and employ an LCM algorithm [15] for tree construction, giving an approximate solution to the Steiner Tree problem.

IV. Integrated Resource Allocation Scheme to Maximize Users' Utility

We formulate the resource allocation as a maximization problem of "utility", which is defined as the ratio of "benefit" to "cost". The benefit forms a monotonically increasing function in the case of video quality. The cost reflects the load on the system and monotonically increases with respect to the amount of allocated resources. With these functions, we can expect high-quality video multicasting without increasing the resource usage.

Utility maximization is performed with respect to the whole system and within each cluster. The maximization of the total utility considers the resource allocation among clusters. The idea behind this is illustrated in Fig. 3. This figure shows a typical example of the relationship between benefit and cost taken from our preceding research work on MPEG-2 video streams [5]. Under such conditions, the total utility can be increased by taking only a portion of the shared resources allocated to rich cluster A, which is receiving a high benefit, and giving it to poor cluster B, which is receiving a low benefit because its allocated resources are insufficient. It is expected that this operation will hardly degrade the benefit of cluster A but greatly increase that of cluster B. As a result of this resource re-allocation, the total utility is expected to increase.

We also consider the interdependence among resources to maximize each cluster's utility. For example, if plenty of bandwidth is allocated to one cluster, its members can receive video data coded at a low compression ratio and avoid complex, heavy decoding. However, the required bandwidth can be decreased if the end systems devote a large amount of CPU resources into coding and decoding tasks and accommodate a highly compressed video stream. In a previous work [5], we derived relationships between the video quality and the required network and end systems resources for MPEG-2 video data. The required bandwidth W Mbps for video transfer, the required amount of CPU resource to code video data at the server, S Mcycle/sec, and the required amount of CPU resource to decode video data at the client, C Mcycle/sec can be estimated from the MPEG-2 coding parameters as:

$$W(R,Q,F,G) \cong 3.1^{\log_4 \frac{R}{640\times480}}(\alpha + \frac{\beta}{Q} - \frac{\gamma}{Q^2})\frac{F}{30}W_{base} \ , \tag{1}$$

$$S \cong S_G \frac{R}{640\times480}\frac{F}{30} \ , \tag{2}$$

$$C \cong W \times 40 + (\lambda + \frac{N_p}{N}\delta + \frac{N_b}{N}\varepsilon) \times \frac{R}{640\times480}\frac{F}{30} \ . \tag{3}$$

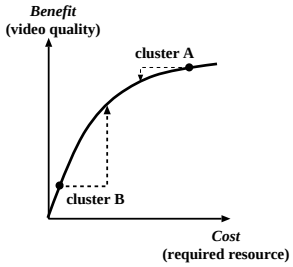

Fig. 3. Relationship between benefit and cost

For details, refer to the paper [5].

By using (1) through (3), given the amount of available resources, we can find an appropriate set of coding parameters to achieve a high-quality video stream with those resources. Thus, A cluster's utility can be maximized by combining (1) through (3) and carefully choosing the coding parameters.

A. Definition of utility

The total utility U is given as the sum of the clusters' utilities U_i, which is defined as a function of the benefit B_i with the allocated resources and the cost P_i paid for them:

$$U = \sum_i^k U_i = \sum_i^k B_i/P_i \ . \tag{4}$$

In this manner, a higher utility can be obtained by providing a higher quality of video to users while keeping the resource utilization lower.

The benefit of cluster i, B_i, is represented as a product of the video quality q_i, which we define as a reciprocal of the quantizer scale Q_i, and the number of clients m_i. Any other definition of q_i fits our scheme as long as it forms a monotonically increasing function with respect to resources:

$$B_i = q_i \times m_i \ . \tag{5}$$

A cluster's cost consists of three different costs, based on the server CPU, the network bandwidth, and the client CPU resources. We define the cost function as follows:

$$P_i = \zeta\{P_i^W\}^2 + \eta\{P_i^S\}^2 + \theta\{P_i^C\}^2 \ , \tag{6}$$

where ζ, η and θ are positive constants that define the importance of each resource. Appropriate determination of these constants is beyond the scope of this paper and remains as a topic for future research work. In our experiments, these constants are all set to one.

The bandwidth cost P_i^W is related to the bandwidth usage of the multicast tree i. Since the bandwidth cost increases as the number of links n_i in the multicast tree increases, the cost is defined as follows:

$$P_i^W = n_i \times \frac{W_i}{W_i^{free}} \ , \tag{7}$$

where W_i is the bandwidth required for the video stream assigned to cluster i, and W_i^{free} indicates the available bandwidth of multicast tree i —that is, the available bandwidth of the tree's bottleneck link.

The server CPU cost P_i^S is defined as the ratio of the amount of resources required for encoding video data to the available amount of server CPU resources S_i^{free} allocated to the cluster i:

$$P_i^S = \frac{S_i}{S_i^{free}} \ . \tag{8}$$

The cluster CPU cost P_i^C is given as the average utilization of the client CPU resource:

$$P_i^C = \frac{1}{m_i} \sum_j \frac{C_i}{C_{ij}^{free}} \ , \tag{9}$$

where C_i and C_{ij}^{free} stand for the amount of CPU resources required for decoding the video data and the available amount of CPU resources for client j of cluster i, respectively.

B. Utility maximization

We formulate resource allocation as a maximization problem of the total utility as defined in Section IV-A. By solving this optimization problem, we can determine an efficient resource allocation for the whole system.

$$maximize \ U \tag{10}$$

$$\begin{aligned}
&\text{Under the constraints} \\
\forall l \quad \textstyle\sum_i^k W_i \cdot Z(i,l) &\leq L_l^{free} && (11)\\
\forall i \quad W_i &\leq W_i^{free} && (12)\\
\forall i \quad S_i &\leq S_i^{free} && (13)\\
\textstyle\sum_i^k S_i^{free} &\leq S^{free} && (14)\\
\forall i,j \quad C_i &\leq C_{ij}^{free} \ , && (15)
\end{aligned}$$

where k and L_l^{free} stand for the number of clusters and the available bandwidth of link l, respectively; And $Z(i,l)$ is one if a multicast tree contains link l, and zero otherwise.

We solve the problem by the following heuristic algorithm:

1) Allocate the server CPU resources equally to each cluster. Bandwidth has already been allocated to the clusters during the tree construction phase.
2) Determine the resource allocation that maximizes the utility of each cluster within the available resources by choosing appropriate set of coding parameters with (1), (2) and (3).
3) Subtract the allocated resources from the system resources.
4) Re-allocate the remaining resources on the server CPU and the links to the cluster whose utility increases the most with the newly allocated resources.
5) Repeat steps 2 through 4 until no cluster can increase its utility or no resources remain.

By using this scheme, the resource allocation can be decided for each cluster and the quality of the video can also be determined from the relationship described in Section IV.

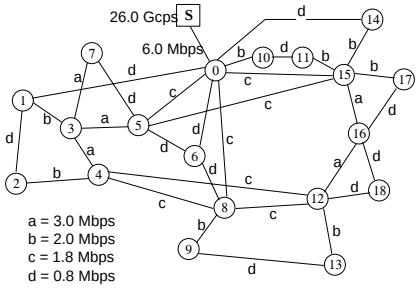

Fig. 4. General-topology network model

V. Evaluation

In this section, we evaluate the effectiveness and appropriateness of our proposed scheme by applying it to a multicast network with heterogenous clients.

A. Simulation model

In our simulation experiments, we use the general-topology network model shown in Fig. 4, which is taken from an MCI network and consists of 19 nodes and 32 internal links. A server is connected to node "0" via a 6.0-Mbps link. Clients are connected to randomly chosen nodes. The bandwidth available for a video session on each access link and the CPU resources available for decoding tasks are chosen at random as long as they can enable a video stream of the minimum quality. In our experiments, all clients are assumed to join the same video session for the sequence "Animation", whose spatial resolution is 160×120 pixels and temporal resolution is 30 fps. This means that only the SNR resolution Q and the GoP structure contribute to resource allocation. In the resource allocation phase, the server determines an appropriate set of coding parameters to maximize cluster utility. The SNR resolution (i.e., the quantizer scale Q_i) ranges from 4 (33.25 dB) to 40 (18.93 dB) at intervals of four. In this paper, based on our previous research work, only the quantizer scale determines the perceived video quality [2].

B. Simulation results

The results of experiments for ten clients in the general-topology network model are shown in Fig. 5. The figure shows the transition of the total utility for all k from one to ten, and the average video qualities in terms of the quantizer scales Q among the clients. A total utility of zero implies that the resource allocation failed. A smaller Q implies a higher-quality video. Fig. 5 clearly shows the trade-off between the number of clusters and the total utility. In these experiments, clusters of one and two failed since some clients had insufficient resources and no set of coding parameters could satisfy all the clients in a cluster. As the number of clusters k increased, the clustering algorithm could group clients into multicast sessions according to their resource availability in more effective and appropriate ways. Then, the resource allocation algorithm successfully determined sets of coding parameters and amounts of resources to allocate to each multicast session. However, beyond $k = 8$ the

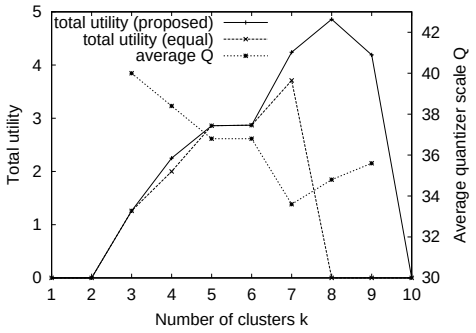

Fig. 5. Transition of utility (10 clients) & average video quality

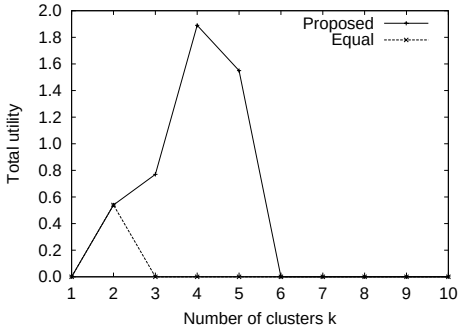

Fig. 6. Transition of utility (100 clients)

total utility began to decrease. This is because a larger k leads to division of the shared resources into small pieces, and thus each of the allocated resources is fully utilized and therefore the cost increases. Although the average quality is the highest in the case of $k = 7$, the utility is not maximum due to the undesirably high cost. We can conclude that constructing eight multicast groups is the best way to efficiently provide clients with video streams when we consider the trade-off between the cost and the benefit.

For comparison purposes, we also conducted resource allocation with only the first two steps of the heuristic algorithm, that is, the server CPU resources were divided equally, the bandwidth was allocated in a max-min fair manner, and utility maximization was carried out within each cluster. This strategy, called "equal resource allocation", can be regarded as a rather conventional simulcast, except that the video quality is determined so as to maximize the cluster's utility within the allocated resources. In the case of equal resource allocation, as shown in Fig. 5, all trials failed except from $k = 3$ to 7. The maximum utility was obtained in the case of $k = 7$, but it was 3.71, far below that obtained with our scheme, and, of course, the quality of the video stream provided to the clients was low.

Fig. 6 shows that our scheme can obtain a feasible resource allocation in a general network with 100 clients. The client resource availabilities and connecting nodes are set at random on the network of Fig. 4. Only four multicast groups were sufficient for 100 heterogeneous clients. In this case, we do not need to try utility maximization beyond $k = 27$ clusters even for more than 100 clients. This number is derived by dividing the server access link bandwidth by the bandwidth required to transfer a video stream of the minimum quality. In practice, for a 100-client session, trying utility maximization more than dozen of clusters does not lead to a better allocation.

VI. Conclusion

In this paper, we have proposed a resource allocation scheme to provide efficient, high-quality video multicast services for heterogeneous clients. The resource allocation is formulated as a utility maximization problem that takes into account the relationships among resources. Several issues still remain. One is that the KA clustering phase does not necessarily lead to better grouping than the random algorithm where initial k points are randomly chosen, although results were not shown due to a limited space. We should consider a new clustering algorithm tied up with our resource allocation scheme. Another problem is related to the practicality of our scheme: it may lack scalability due to the fact that the algorithms are designed for centralized control.

Acknowledgments

This work was partly supported by Special Coordination Funds for promoting Science and Technology of the Ministry of Education, Culture, Sports, Science and Technology of Japan, and Telecommunication Advancement Organization of Japan.

References

[1] K. Lakshman and R. Yavatkar, "Integrated CPU and network-I/O QoS management in an endsystem," in *Proceedings of IFIP IWQoS '97*, pp. 167–178, May 1997.

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

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

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

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

[6] K. Fukuda, *Integrated QoS Control Mechanisms for Real-Time Multimedia Systems in Reservation-Based Networks*. PhD thesis, Osaka University, January 2000.

[7] H. Tokuda, T. Nakajima, and P. Rao, "Real-Time Mach: Towards a predictable Real-Time system," in *Proceedings of USENIX Mach Workshop*, pp. 73–82, October 1990.

[8] M. Handley and V. Jacobson, "SDP: Session description protocol," *RFC2327*, April 1998.

[9] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *Proceedings of ACM SIGCOMM '96*, pp. 117–130, September 1996.

[10] J. Smith, K. Calvert, S. Murphy, H. Orman, and L. Peterson, "Activating networks: A progress report," *IEEE Computer*, vol. 32, pp. 32–41, April 1999.

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

[12] J. M. Peña, J. A. Lozano, and P. Larrañaga, "An empirical comparison of four initialization methods for the k–means algorithm," *Pattern Recognition Letters 20*, pp. 1027–1040, July 1999.

[13] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data. An Introduction to Cluster Analysis*. John Wiley & Sons, Inc., 1990.

[14] B. Wang and J. C. Hou, "Multicast routing and its QoS extension: Problems, algorithms, and protocols," *IEEE Network*, January 2000.

[15] A. Shaikh, S. Lu, and K. Shin, "Localized multicast routing," in *Proceedings of IEEE GLOBECOM '95*, pp. 1352–1356, November 1995.