Layered Resource Allocation for Video Broadcasts over Wireless Networks

Junu Kim, Jinsung Cho, *Member*, IEEE, and Heonshik Shin, *Member*, IEEE

Abstract — *This paper aims to combine adaptive modulation and coding with layered video coding to improve the quality of video services to users experiencing differing radio conditions, in the context of broadcast and multicast standards such as MBMS and BCMCS. We propose an optimal radio resource allocation algorithm which maximizes a general performance metric for a video session in polynomial time. We show that system-wide optimal resource allocation can be obtained by combining our algorithm with a simple two-step decomposition of the system. In some configurations frequent re-allocations of resource are required, so we also present a sub-optimal allocation algorithm which runs in near linear time. Simulation results show better video quality than existing resource allocation schemes over a range of conditions, and also suggest that the difference between the performance of optimal and sub-optimal solutions is less than 3%*[1].

Index Terms — radio resource allocation, wireless video broadcast, layered video coding, adaptive modulation and coding.

I. INTRODUCTION

Recently, the 3GPP and 3GPP2 groups have specified resource-efficient ways of delivering broadcast and multicast services: multimedia broadcast and multicast service (MBMS) [1], and broadcast and multicast services (BCMCS) [2], respectively. Since these standards have many things in common, we use the term "broadcast standards" to refer to both of them.

These new broadcast standards specify a shared radio link implemented by a low bit-rate single modulation and coding scheme (MCS) to provide full area coverage regardless of users' radio conditions. This contrasts with unicast services, in which adaptive modulation and coding (AMC) [3] selects MCS level that gives the best bit-rate for the radio condition at each receiver. Using AMC in broadcast and multicast services can allow increased throughput for those users who experience good radio conditions without much modification of user devices. However, this leads us to an open problem; how should radio resources be allocated to different MCS levels when users are experiencing a wide range of radio conditions. We also need to consider how to adapt the transmission of video data to the resulting radio resource allocation.

[1] This work was supported in part by the Brain Korea 21 Project. The ICT at Seoul National University provided research facilities for this study.

Junu Kim and Heonshik Shin are with the School of Computer Science and Engineering, Seoul National University, Seoul, Korea (e-mail: junukim@cslab.snu.ac.kr, shinhs@snu.ac.kr).

Jinsung Cho is with the Dept. of Computer Engineering, Kyung Hee University, Youngin 446-701, Korea (e-mail: chojs@khu.ac.kr).

There has been a lot of research on combining MCS with video adaptation [4]–[9]. Zhang et al. [4] used multiple description coding [10] as the basis of a dynamic multi-level resource allocation scheme which maximizes throughput in a best-effort manner. But most approaches to this problem have been based on layered video coding [5]–[9]. Initially this involved static multi-layer resource allocation schemes [5], [6]. Some authors used as few as two MCS levels [7], [8] to accommodate the dynamic distribution of users' radio conditions, where as Atici et al. [9] proposed a multi-level compromise between coverage and throughput by dividing resources in the code domain while using a single MCS level. But, the potential of any scheme that fails to exploit multilevel AMC and layered coding effectively is limited. An additional problem with these approaches is their reliance on throughput as a performance metric, instead of the actual quality of the received video.

In this paper, we propose a new layered resource allocation which fully utilizes the AMC and layered coding facilities in broadcast standards. Adapting a generalized performance metric to accommodate various quality measures, we will describe an algorithm which maximizes the quality of each video session, while guaranteeing a minimum video quality to all users; and this algorithm runs in polynomial time under any given resource budget. We show that a system-wide optimal resource allocation can be obtained with a simple two-step decomposition of allocation that deals with each video session and then with the system as a whole. We also present a sub-optimal system-wide allocation algorithm of reduced computational complexity, which is useful when the distribution of users' radio conditions changes frequently.

The rest of this paper is organized as follows: in Section II, we describe the model of our layered resource allocation scheme and explain several fundamental concepts. We formulate the resource allocation problem in Section III. In Section IV we present an optimal allocation algorithm with polynomial time complexity, and in Section V we present the sub-optimal algorithm. Section VI is an evaluation of these algorithms and Section VII concludes this paper.

II. SYSTEM MODEL AND FUNDAMENTAL CONCEPTS INCLUDING UTILITY

A. System Model

Recent broadcast standards have specified a new type of control point for their video services: this is called a broadcast/multicast service center in MBMS, and a BCMCS controller in BCMCS. These control points are responsible for

Contributed Paper
Manuscript received October 12, 2008

providing broadcast and multicast services and for announcing these services to users. There is also a node which controls the allocation of radio resources in a cell, such as the base station controller in the CDMA2000 system [11], and we assume that this node cooperates with the control point to run our resource allocation scheme. For simplicity, we will ignore unicast voice or data sessions and focus on broadcast and multicast video sessions in a single cell. However, since our layered resource allocation scheme dynamically adapts to the given resource budget, it can also be used with unicast sessions.

Our resource allocation scheme can use any of the popular layered coding, such as MPEG-4 fine granularity scalability (FGS) coding [12] or H.264/AVC scalable video coding [13]. In layered coding, a raw video is encoded into multiple layers: a base layer provides coarse video and additional enhancement layers provide incremental improvements. Users receive as many additional enhancement layers as their resource permit. In advanced layered coding schemes, the creation of a layered video stream can be separated from resource allocation. This allows the computation-intensive operations required for video coding to be performed outside the system, very likely by the content provider.

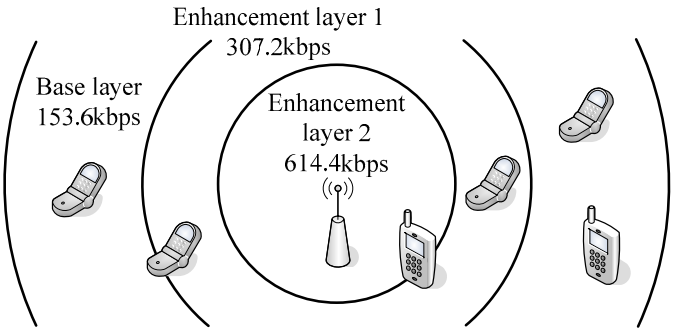

Fig. 1. Conceptual layered resource allocation. The concentric circles represent the coverage of each video layer.

Our layered resource allocation scheme is illustrated in Fig. 1. MCS levels are arranged in order of bit-rate: a lower level has a lower bit-rate and a larger coverage area than a higher level. Users of the higher levels can also receive video data from lower levels. The base layer with the lowest bit-rate is represented by the largest circle. The MCS levels with the greatest coverage contain the more important video data.

We consider various models of user radio conditions. In broadcast services where no feedback from users is allowed, a predefined model such as a uniform or normal distribution has to be used. But, if the broadcast system can collect performance feedback from users, then it can build a statistical model. Frequent changes to these models are usually unnecessary, so that resource allocation is mainly performed at the beginning or end of a video session. This permits the use of a computationally intensive optimal allocation algorithm. But in multicast the amount of feedback from users is limited and resource allocation can adapt dynamically to user radio conditions. In this case, frequent resource adaptation required, making a fast sub-optimal algorithm more suitable.

B. Fundamental Concepts

Broadcast standards employ a time-shared forward link. The fundamental timing unit for this link is called a timeslot, which identifies the pilot and MAC channels, and it also has a data portion that may contain traffic or a control channel. However, the amount of data carried by a timeslot cannot be fixed under AMC. So we define a fixed-size data unit, or block, which carries a fixed amount of the encoded video data: when the data portion of a timeslot is smaller than a block, several timeslots are required to carry a block; but when the data portion of a timeslot is larger than a block, more than one block can be carried in a timeslot. The size of a block is determined to suit the system and the video coding.

There is no consensus on video performance metrics. A simple throughput metric is widely used [4], [7]–[9], but receiver-based absolute performance metrics such as perceived video quality [14] are more clearly focused on users. Relative performance metrics [15] may be a better way to represent the extent to which users' expectations of a video service are satisfied. A practical choice of performance metric depends on a number of factors, but we assume a generic performance metric which we call utility; and we use the term a system utility, to denote the total quality of a broadcast system, defined as the sum of the utilities across all the users.

III. PROBLEM DESCRIPTION

Let us first consider encoding a video session into multiple layers and mapping each video layer to a suitable MCS level. The resource allocated to a video session s under an n-timeslot budget can be represented by a block allocation vector $V_s(n)$ as follows:

$$V_s(n) = \{b_s^1, \cdots, b_s^k, \cdots, b_s^L\}, (1 \le k \le L, \beta_s \le b_s^k \le B_s),$$

where L is the total number of MCS levels in the system, and b_s^k is the number of cumulative blocks available in the k th level during a given scheduling period. A user who subscribes to the k th level receives timeslots from the 1st to the k th level, so that they receive total b_s^k blocks. The minimum and maximum number of blocks required for the video session during a given scheduling period is β_s, and B_s, respectively.

The optimal utility of a video session s under an n-timeslot budget, denoted by $U_s(n)$, can be defined as the maximum value of the sum of the utilities for all the users:

$$U_s(n) = \max_{\text{for all } l_s, b_s^k} \{\sum_{k=1}^{L} W_s^k u_s(n_s, l_s, k, b_s^k)\}, \quad (1)$$

where l_s is the total number of used MCS levels and W_s^k is the proportion of users of video session s who can receive levels up to and including the k th. The utility to the user who subscribes to a session s with the k th level under an N-timeslot budget is defined to be $u_s(n_s, l_s, k, b_s^k)$.

The optimal system utility under an N -timeslot budget with S video sessions, denoted by $U(N,S)$ is defined as the sum of the prioritized utilities of each video session:

$$U(N,S)=\max_{\substack{0<\sum_{s=1}^{S} n_s \le N \\ \text{for all } l_s, b_s^k}} \{\sum_{s=1}^{S} P_s \sum_{k=1}^{L} W_s^k u_s(n_s,l_s,k,b_s^k)\},$$

where P_s is the user preference for video session S, and n_s is the timeslot budget for each session.

IV. Optimal Resource Allocation

A. Layered Resource Allocation in a Video Session

We are going to build a recurrence relation that describes the process of our layered resource allocation (LRA) algorithm for a video session. We start by defining the condition $C_s(n,l,k,b)$ for a video session s: under an n-timeslot budget, the total number of used MCS levels is l, and users who can access the k th level receive b blocks. Under condition $C_s(n,l,k,b)$, b blocks are assigned to l levels selected from the 1st to the $(k-1)$ th, while no blocks are assigned to higher levels; or $(b-i)$ blocks are allocated to $(l-1)$ levels chosen from the 1st to the $(k-1)$ th, while i blocks are assigned to the k th level. Therefore, the condition $C_s(n,l,k,b)$ can only be satisfied in the form $C_s(n,l,k-1,b)$ or $C_s(n-\lceil T^k i \rceil, l-1, k-1, b-i)$ for $i=1,2,\ldots,b-1$, as shown in Fig. 2. T^k is the number of timeslots that comprises a block in the k th level. If the timeslot budget for the k th level is n and i blocks are allocated to the k th level, then the timeslot budget for the $(k-1)$ th level is $n-\lceil T^k i \rceil$.

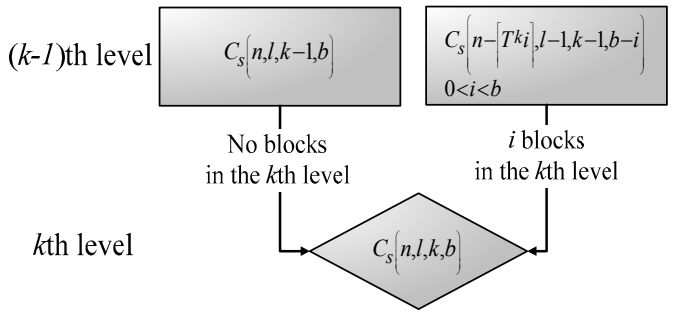

Fig. 2. A recurrence relation of C_s between the k th and $(k-1)$ th levels.

Defining the optimal utility of the video session s under $C_s(n,l,k,b)$ as $f_s(n,l,k,b)$, we can obtain $U_s(n)$ from the maximum value of the function $f_s(n,l,k,b)$. The proof is shown in Appendix A. To solve f_s, we introduce several auxiliary functions. First, we define $\Gamma_s(n,l,k,b,i)$ to be the optimal utility of a video session s when i blocks are allocated to the k th level under the condition $C_s(n-\lceil T^k i \rceil, l-1, k-1, b-i)$. If we allocate i blocks to the k th level when $(b-i)$ blocks are already allocated to the $(k-1)$ th and lower levels, then users who can access the k th level can benefit from this additional allocation, but it makes no difference to users who only have access to lower levels. Therefore $\Gamma_s(n,l,k,b,i)$ can be formulated as the sum of the optimal session utility of the $(k-1)$ th level and the change to the session utility, denoted by $\Delta_s(n,l,k,b,i)$, that results from adding i blocks to the k th level, as follows:

$$\begin{aligned}\Gamma_s(n,l,k,b,i) &= \Delta_s(n,l,k,b,i) \\ &\quad + f_s(n-\lceil T^k i \rceil, l-1, k-1, b-i).\end{aligned} \tag{2}$$

To obtain an expression for $\Delta_s(n,l,k,b,i)$, we introduce another auxiliary function $g_s(n,l,k,b)$, which is the sum of the weighted utilities experienced by users who can access to the k th levels under the condition $C_s(n,l,k,b)$. If we allocate i blocks to the k th level, then users who can access the k th receive new utilities, $g_s(n,l,k,b)$, which are greater than their old utilities, $g(n,l-1,k,b-i)$. This change can be expressed as follows:

$$\Delta_s(n,l,k,b,i)=g_s(n,l,k,b)-g_s(n,l-1,k,b-i), \tag{3}$$

where $g_s(n,l,k,b)$

$$=\begin{cases}\sum_{i=k}^{L} W_s^i u_s(n,l,i,b) & (C_s(n,l,k,b) \text{ is satisfied}) \\ 0 & (\text{otherwise}).\end{cases} \tag{4}$$

We can calculate $u_s(n,l,i,b)$ from the pre-encoded video stream. We can calculate g_s and Δ_s from these equations using the distribution of users' radio conditions.

We now can formulate f_s using the recurrence relation and equations above. If blocks are allocated to the k th level, then $f_s(n,l,k,b)$ is the maximum value of $\Gamma_s(n,l,k,b,i)$, from the definition of f_s. Otherwise we skip the k th level,

which does not affect the users, so that $f_s(n,l,k,b) = f_s(n,l,k-1,b)$. If $l=1$, then all blocks are allocated to a single MCS level and we skip the other levels, so that $f_s(n,1,k,b) = g_s(n,1,1,b)$ for all n, k and b; this follows from the definition of f_s and g_s. We can now formulate $f_s(n,l,k,b)$ as follows:

$$f_s(n,l,k,b) = \begin{cases} \max\left\{ \begin{matrix} \max\limits_{0\le i<b}\{\Gamma(n,l,k,b,i)\}, \\ f_s(n,l,k-1,b) \end{matrix} \right\} & (l \neq 1) \\ g_s(n,l,k,b) & (l=1), \end{cases} \quad (5)$$

if $C_s(n,l,k,b)$ is not satisfied, $f_s(n,l,k,b) = 0$.

These recurrence relations can be solved by dynamic programming using a bottom-up approach. Initially, $f_s(n,1,k,b)$ is calculated for all n, k and b. Values of $f_s(n,l,k,b)$ are then calculated incrementing l by 1 in each case, and using all the previous values of f_s. This step is repeated until l reaches its maximum value of L. The maximum value of $f_s(n,l,k,b)$ is $U_s(n)$. Since $1 \le l \le L, 1 \le k \le L$ and $1 \le b \le \lceil n/T^L \rceil$, the maximum number of different instances of $f_s(n,l,k,b)$ is $O(n^2 \cdot L^2)$.

B. System-wide Radio Resource Allocation

A two-step decomposition technique [16] makes it possible to allocate radio resources separately to the video sessions and across the system. If there are N timeslots in the system, the optimal system utility is the maximum value of the sum of all the optimal session utilities. That is,

$$U(N,S) = \max_{0<\sum_{s=1}^{S} n_s \le N} \{\sum_{s=1}^{S} P_s U_s(n_s)\}.$$

Since LRA produces $U_s(n)$ for $1 < n \le N$ and $1 < s \le S$, we can obtain $U(N,S)$. The proof is given in Appendix B. Appendix C shows how we can calculate $U(N,S)$ using a recurrence relation in $O(N^2 \cdot S)$ times.

C. Complexity of LRA

We can calculate $u_s(n,l,k,b)$ from the video stream offline. Knowing u_s, we can solve g_s in $O(L)$ time, and therefore Δ_s can also be calculated in $O(L)$ time. Using (2), (3) and (5), we can solve $f_s(n,l,k,b)$ in $O(n^3 \cdot L^3)$ time, since the space required for f_s is $O(n^2 \cdot L^2)$ and $1 \le i < b$, as shown in Section IV. A. Therefore we can conclude that the computational complexity of $U_s(N)$ is $O(N^3 \cdot L^3)$, based on Theorem 1 in Appendix A. This result shows that we can solve the optimal resource allocation problem in polynomial time.

V. SUB-OPTIMAL RESOURCE ALLOCATION

A. The Sub-optimal Algorithm

The sub-optimal radio resource allocation (sLRA) algorithm performs system-wide resource allocation more quickly than LRA, making it more suitable when resources must be frequently be reallocated. The sLRA algorithm allocates a block to a certain MCS level of a video session at each iteration, with the aim of increasing the potential system utility. The potential system utility is defined as the weighted sum of the potential utilities of all receivers. A receiver achieves potential utility when all the remaining timeslots in the system are used to maximize the utility that it experiences. For example, $N-r$ timeslots have been allocated to the system, the utility of a receiver is maximized if all the remaining r timeslots are assigned to its highest receivable MCS level.

Algorithm sLRA

$r \Leftarrow N$
for $i = 1$ to S **do**
  $V_i(N) \Leftarrow 0$ and $b_i^{\lambda_i} \Leftarrow \beta_i$
  $r \Leftarrow r - \lceil T^{\lambda_i} \cdot \beta_i \rceil$
end for
while $r > T^L$ **do**
  for $i = 1$ to S and $j = 1$ to L **do**
    calculate $v(r,i,j)$
  end for
  $(x,y) \Leftarrow (i,j)$ which maximizes $v(r,i,j)$
  for $j = y$ to L **do**
    $b_x^j \Leftarrow b_x^j + 1$
  end for
  $r \Leftarrow r - \lceil T^y \rceil$
end while

Fig. 3. The pseudocode of sLRA algorithm.

Fig. 3 shows the pseudocode of sLRA. Initially, sLRA allocates j blocks to the lowest MCS level of the j th session, denoted by λ_j, to meet the minimum video quality. Next it selects the x th level of the video session y, which maximizes the potential system utility $v(r,x,y)$ using the remaining r timeslots from the timeslot budget of N. After a level has been chosen, the allocation vector $V_x(n)$ and the remaining r timeslots are updated. This procedure is repeated until no more blocks can be allocated.

To calculate $\upsilon(r,i,j)$, we need to consider the weighted sum of the potential utilities of the receivers, denoted by $h_i(r,j)$, which can receive up to the j th MCS level of video session i, using r timeslots remaining in the timeslot budget of N. This weighted sum is defined as follows:

$$h_i(r,j) = W_i^j u_i(N,l,j,\lceil r/T^j \rceil + b_i^j).$$

When a block is allocated to the x th level of video session y, that block is received by all subscribers to the session y who are able to receive the x th level; and the block is no use to subscribers to session y who are only able to receive up to the $(x-1)$ th level, and users who are not subscribing to the session y at all. Therefore $\upsilon(r,x,y)$ can be expressed as follows:

$$\upsilon(r,x,y) = \sum_{i=1}^{S} P_i \sum_{j=1}^{L} h_i(r - \lceil r/T^y \rceil, j) + \sum_{j=y}^{L} P_x \{h_x(r,j) - h_x(r - \lceil r/T^y \rceil, j)\}.$$

B. Complexity of the Sub-optimal Algorithm

The number of iterations of the while loop varies because as the size of T^j changes: However it is always less than or equal to $\lceil n/T^L \rceil$, since $T^L < T^j$ for $1 \le j < L$. The time complexity of calculating $C(r,i,j)$ is $\mathrm{O}(S \cdot L \cdot E)$, and the complexity of determining $h_i(r,j)$ is $\mathrm{O}(E)$. We showed that $u_s(n,l,k,b)$ can be solved offline in Section IV.C: thus $h_i(r,j)$ can be determined with constant time complexity. Therefore we can solve the inner 'for' loop which calculates $C(r,i,j)$ in $\mathrm{O}(S^2 \cdot L^2)$ time. Therefore the time-complexity of system-wide radio resource allocation using sLRA does not exceed $\mathrm{O}(\lceil N/T^k \rceil \cdot S^2 \cdot L^2)$. In a given system, L and T^L are constant, and S only changes at the beginning or end of a video session. Therefore, in most cases, sLRA can perform radio resource allocation in linear time ($\mathrm{O}(N)$).

VI. EVALUATION

In this section we analyze the methods of resource allocation that we have proposed, using simulation techniques, and compare our approach with other allocation algorithms.

A. Simulation Methodology

We evaluated our resource allocation algorithm within a simulated CDMA2000 1xEV-DO Rev. A cellular network. We assumed that the length of a scheduling period is 1 second and that the number of timeslots in the system, N, is 600, because 1xEV-DO specifies that the length of a timeslot is 1.667ms. The configuration of MCS level is based on the data-rate control steps defined in the 1xEV-DO specification. We assume that there are seven levels with data-rates between 153.6kbps and 2457.6kbps. Agashe et al. [11] showed that a bit-rate of 153.6kbps can be supported using the Reed-Solomon forward error-correction code and combined radio links in more than 90% of a BCMCS coverage area. In our simulation we have assumed that the size of a block is 12kbits, a figure that we obtained by considering the bit-rates of the different MCS levels. Each value of T^k is calculated for the corresponding MCS level.

The MoMuSys reference MPEG-4 FGS codec [17] is used as the encoder to generate bit-streams. Since this scalable codec is based on bit-planes, it generates a base layer, and an enhancement layer which can be partitioned arbitrarily to make multiple additional enhancement layers. We used the popular Foreman QCIF 15Hz sequence in the simulation. The bit-rate of the encoded bit-streams ranged from 36kbps to 144kbps. Since the size of a block is 12kbits, encoded streams require from 3 to 12 blocks per second, and thus $\beta_s = 3$ and $B_s = 12$.

We assume that users are randomly distributed in a cell, which means that variations in users' radio conditions can be modeled as a distribution. The number of video sessions is different in each experiment.

We adopted peak-to-noise-ratio (PSNR), which is a widely used measure of video quality [17], as the utility function in our simulation. A PSNR value in decibels (dB) can be computed as follows:

$$PSNR = 20\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right),$$

$$MSE = \frac{\sum (f(i,j) - F(i,j))^2}{WIDTH \times HEIGHT},$$

where MAX_I is the maximum possible pixel value in the image, $f(i,j)$ is a source image of $WIDTH \times HEIGHT$ pixels, and $F(i,j)$ is an image reconstructed from the encoded source image. The PSNR of a video session can be calculated by averaging the PSNR of each frame in the session. The utility is the PSNR value perceived by a user who subscribes to a video session with a certain MCS.

B. Simulation Results

1) Layered Resource Allocation in a Video Session: First, we evaluated the performance of LRA and sLRA by comparing them with: (a) the existing broadcast standards which use a single MCS level for full area coverage (which we will call the SINGLE algorithm); (b) an adaptive modulation which uses two MCS levels (which we will call AM) [7]; and (c) a uniform allocation of blocks to two MCS levels, one for bad and one for good radio conditions (which we will call UNIFORM) [8]. In UNIFORM, a MCS level

which covers at least 60% of users is selected for an additional enhancement layer. Timeslots are distributed between the base and an enhancement layer so that each layer has the same number of blocks. AM adaptively selects two MCS levels which maximize the users' aggregate bit-rate, assuming that $B_s/2$ blocks are already assigned to the base layer.

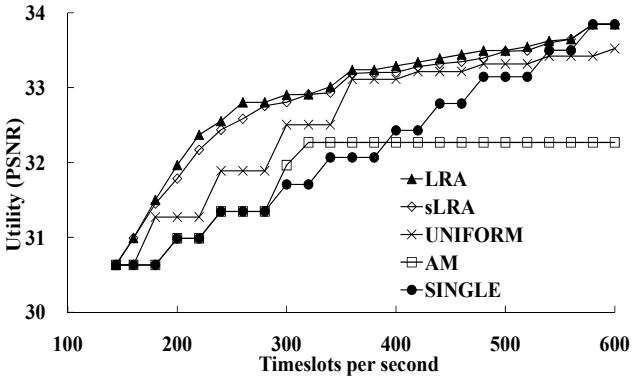

Fig. 4. Session utility against timeslots per second for different allocation schemes.

Fig. 4 shows the resource allocations in a single-session environment as a function of the number of timeslots per second allocated to the session. It is clear that LRA and sLRA outperform all the other schemes under any timeslot budget. The difference between LRA and sLRA is less than 3% in the session shown in this figure. The performance of SINGLE approaches that of LRA as the timeslot budget increases because SINGLE represents the optimal allocation when $n = N$, since $B_s \le \lceil n/T^1 \rceil$ and all B_s blocks can be allocated to the base layer.

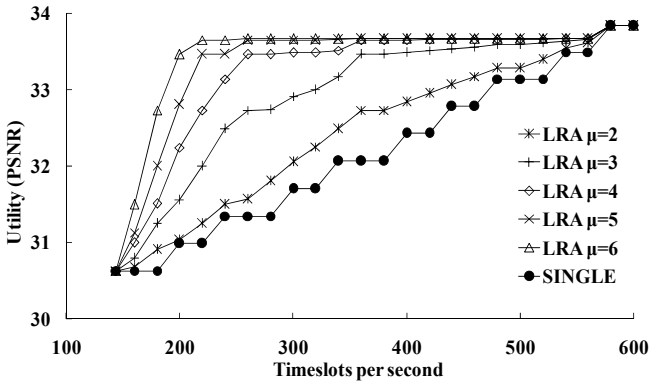

Fig. 5. Session utility achieved by LRA and SINGLE against timeslots per second for different user radio conditions.

Fig. 5 shows the session utility achieved by LRA and SINGLE in a single-session environment as a function of the number of timeslots per second allocated to the session, over a range of user radio conditions. In this experiment we assumed that the number of users subscribing to each MCS level follows a normal distribution with mean μ and unit standard deviation. When $\mu = 2$, so that half of the users subscribe to and including the second level, LRA achieves only slightly higher utility than SINGLE. As μ increases, the utility achievement by LRA grows rapidly, although the maximum session utility is bounded by B_s . These results show how LRA maximizes session utility by adapting to the radio conditions experienced by users.

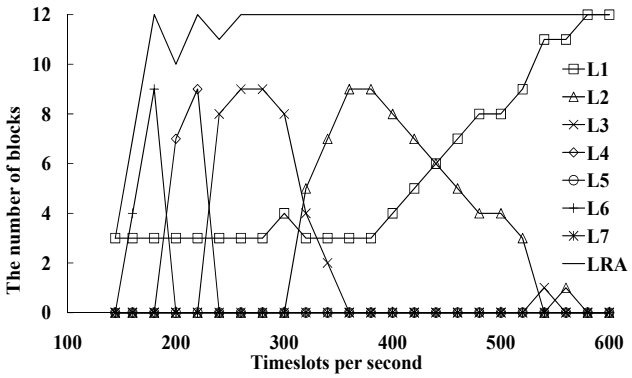

Fig. 6. The number of blocks allocated to each MCS level in a single-session environment and the total number of blocks against the number of timeslots per second.

Next, we consider the way in which LRA manages MCS levels. Fig. 6 shows the number of blocks allocated to each MCS level in a video session as a function of the number of timeslots per second. We can see that LRA dynamically allocates blocks to levels as the timeslot budget increases. At the start, blocks are allocated to the lowest level to guarantee the minimum utility. As the timeslot budget grows, blocks are initially assigned to higher levels, and only later to lower levels. That is because a block in one of the higher levels requires fewer timeslots than a block in one of the lower levels. LRA uses as many blocks as possible, up to B_s , within the given timeslot budget, so as to maximize the session utility.

2) System Utility: We now consider the performance of LRA and sLRA with multiple video sessions. Users can have preferences for different sessions, which can generally be modeled quite well as a Zipf distribution [17]. This can be written as follows:

$$p_j = \frac{(1/j)^\theta}{\sum_{j=1}^{S}(1/j)^\theta} \quad \text{for } j = 1, 2, \ldots, S,$$

where θ is the skew factor. When $\theta = 0$, the Zipf distribution becomes the uniform distribution with $p_j = 1/s$, for all j . As θ increases, the probabilities are skewed. We can model the preferences of users for particular video sessions by increasing θ .

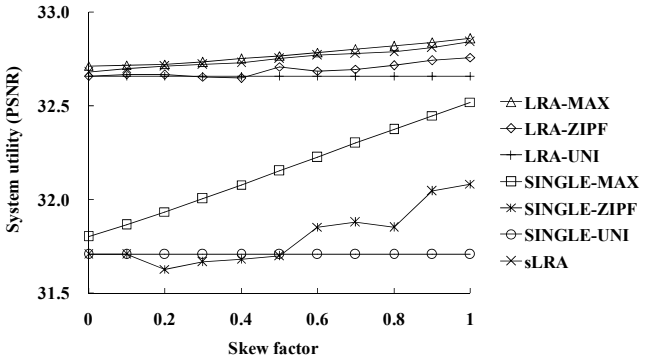

Fig. 7. System utility for a system with four video sessions as a function of the skew factor in the Zipf distribution.

In this simulation, we consider how LRA and SINGLE allocate resources, when they are combined with three different algorithms to allocate resources between sessions: MAX, in which timeslots are optimally distributed as described in Section IV.B; ZIPF, in which the timeslot allocation is directly proportional to user preferences; and UNI, which allocates the same number of timeslots to each video session. We also consider the system-wide resource allocation performed by sLRA. All resource allocations were tested in an environment in which there are four video session. Fig. 7 shows system utility plotted against skew factor.

Clearly, LRA-MAX and sLRA outperform all the other combinations across the full spectrum of skew factors. The difference between LRA-MAX and sLRA is less than 2%. Higher skew factors mean that there are stronger user preferences for some videos than others. In this situation the performance of LRA-MAX and SINGLE-MAX improves, because they assigns more timeslots to the favorite video sessions; but the performance of LRA-UNI and SINGLE-UNI is straight. LRA-ZIPF and SINGLE-ZIPF also perform better as the skew factor increases, but they perform even worse than LRA-UNI and SINGLE-UNI under low skew factors, since their allocation is never optimal.

VII. CONCLUSIONS

We have proposed an optimal layered resource allocation algorithm which maximizes system utility. This work extends existing broadcast and multicast services by the introduction of the layered resource allocation, which enables fine-grained control over coverage and video quality when many users are experiencing heterogeneous radio conditions. Our optimal allocation algorithm runs in polynomial time, making it suitable for practical video services, and maximizes a general utility function, so that alternative formulation of utility can be adapted to meet different circumstances. We also presented a sub-optimal algorithm which performs system-wide resource allocation in near-linear time. Simulations show that the difference in performance between the optimal and the sub-optimal algorithm is less than 3%, and they both outperform existing resource allocation algorithms based on multiple modulation and coding schemes, as well as the allocation methods specified in the broadcast standards. We suggest that this approach is promising to enhancing consumer satisfaction with existing broadcast and multicast video services.

APPENDIX A

PROOF OF $U_s(n) = \max_{i,j \in C_s(n,i,L,j)} \{f_s(n,i,L,j)\}$

Theorem 1: For a video session s which has an n-timeslot budget, the optimal session utility is the maximum value of the function $f_s(n,l,k,b)$ under $C_s(n,l,k,b)$. That is, $U_s(n) = \max_{i,j \in C_s(n,i,L,j)} \{f_s(n,i,L,j)\}$.

Proof: From the definition of $U_s(n)$, we have $U_s(n) \geq \max_{i,j \in C_s(n,i,L,j)} \{f_s(n,i,L,j)\}$. If $V_s(n)$ satisfies the condition $C_s(n,i,L,j)$, then there exists at least one $f_s(n,\hat{i},L,\hat{j})$, where $f_s(n,\hat{i},L,\hat{j}) = U_s(n)$. Since $f_s(n,\hat{i},L,\hat{j}) \leq \max_{i,j \in C_s(n,i,L,j)} \{f_s(n,i,L,j)\}$, we see that $U_s(n) \leq \max_{i,j \in C_s(n,i,L,j)} \{f_s(n,i,L,j)\}$. The theorem is proved by combining the two inequalities.

APPENDIX B

PROOF OF TWO-STEP DECOMPOSITION OF RESOURCE ALLOCATION

Theorem 2: If there are N timeslots in a system, then the optimal system utility is the maximum value of the sum of all the optimal session utilities. That is,

$$U(N,S) = \max_{0 < \sum_{s=1}^{S} n_s \leq N} \{\sum_{s=1}^{S} P_s U_s(n_s)\}.$$

Proof: From the definition of $U(N,S)$, it is clear that $U(N,S) \geq \max_{0 < \sum_{s=1}^{S} n_s \leq N} \{\sum_{s=1}^{S} P_s U_s(n_s)\}$. If an optimal utility $U(N,S)$ exists, then there is a partition of timeslots $\{\hat{n}_1, \hat{n}_1, \ldots, \hat{n}_S\}$ among video sessions, and $U(N,S) = \sum_{s=1}^{S} P_s U_s(\hat{n}_s) \leq \max_{0 < \sum_{s=1}^{S} n_s \leq N} \{\sum_{s=1}^{S} P_s U_s(n_s)\}$. These two inequalities prove our theorem.

APPENDIX C
SYSTEM-WIDE RADIO RESOURCE ALLOCATION

Theorem 3: When there is an N-timeslot budget and S video sessions, the optimal system utility $U(N,S)$ has the following recurrence relations:

for $0 \le n \le N, 0 \le s \le S$,

if $n \cdot s = 0$ then $U(n,s) = 0$,

else $U(n,s) = \max_{1<i\le n}\{P_s U_s(i) + U(n-i, s-1)\}$.

Proof: The theorem can be proved by induction. We start with the basic case of $s = 1$:

$$U(n,1) = \max_{1<i\le n}\{P_1 U_1(i) + U(n-i,0)\} = \max_{1<i\le n}\{P_1 U_1(i)\}.$$

This equation is true from the definition of $U(N,S)$. We now assume that, when $s = k$, the utility $U(N,k)$ obeys the above recurrence relations, so that

$$U(n,k+1) = \max_{1<i\le n}\{P_{k+1} U_{k+1}(i) + U(n-i,k)\} = P_{k+1} U_{k+1}(\hat{i}) + U(n-\hat{i},k).$$

Theorem 2 shows that $U_s(i)$ depends only on the timeslot budget i, and is independent of other video sessions. Therefore there exists a value of $\hat{i}$ which partitions the timeslot budget between the $(k+1)$ th session and the other sessions, while maximizing $U(n,k+1)$.

REFERENCES

[1] *Multimedia Broadcast/Multicast Service (MBMS): architecture and functional description*, 3GPP Std. TS 23.246, 2005.

[2] P. Agashe, R. Rezaiifar, P. Bender, and QUALCOMM, "CDMA2000 high rate broadcast packet data air interface design," *IEEE Commun. Mag*., vol. 42, no. 2, pp. 83–89, Feb. 2004.

[3] A. J. Goldsmith and P. P. Varaiya, "Capacity of fading channels with channel side information," *IEEE Trans. Inf. Theory*, vol. 43, pp. 1986–1992, Nov. 1997.

[4] X. Zhang and Q. Du, "Cross-layer modeling for QoS-driven multimedia multicast/broadcast over fading channels in mobile wireless networks," *IEEE Commun. Mag*., pp. 62–70, Aug. 2007.

[5] Z. Cakareski, N. Ahmed, A. Dhar, and B. Aazhang, "Multilevel coding of broadcast video over wireless channels," in *Proc. IEEE ICASSP*, Orlando, FL, USA, May 2002, pp. 13–17.

[6] M. V. der Schaar and J. Meehan, "Robust transmission of MPEG-IV scalable video over 4G wireless networks," in *Proc. IEEE ICIP*, Rochester, NY, USA, Jun. 2002, pp. 24–28.

[7] C. Hwang and Y. Kim, "An adaptive modulation method for multicast communication of hierarchical data in wireless network," in *Proc. IEEE ICC*, Apr. 2002, pp. 896–900.

[8] A. M. C. Correia, J. C. M. Silva, N. M. B. Souto, L. A. C. Silva, A. B. Boal, and A. B. Soares, "Multi-resolution broadcast/multicast systems for MBMS," *IEEE Trans. Broadcast*., vol. 53, no. 1, pp. 224–234, Mar. 2007.

[9] C. Atici and M. O. Sunay, "High data-rate video broadcasting over 3G wireless systems," *IEEE Trans. Broadcast*., vol. 53, no. 1, pp. 212–223, Mar. 2007.

[10] V. K. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Process. Mag*., vol. 18, no. 5, pp. 74–94, 2001.

[11] *CDMA2000 high rate packet data air interface specification*, 3GPP2 Std. C.S0024-A, 2004.

[12] W. Li, "Overview of fine granularity scalability of MPEG-4 video standard," *IEEE Trans. Circuits Syst. Video Technol*., vol. 11, no. 3, pp. 301–317, Mar. 2001.

[13] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[14] G. Bianchi, A. T. Campbell, and R. F. Liao, "On utility-fair adaptive services in wireless networks," in *Proc. Int. 6th Workshop Quality of Services*, May 1998.

[15] J. Liu, B. Li, and Y. Q.Zhang, "Optimal partitioning of multicast receivers," *IEEE Trans. Multimedia*, vol. 6, pp. 87–102, Feb. 2004.

[16] J. Liu, B. Li, Y. T. Hou, and I. Chlamtac, "On optimal layering and bandwidth allocation for multi-session video broadcasting," *IEEE Trans. Wireless Commun.*, Feb. 2004.

[17] A. Pearmain, A. Carvalho, A. Hamosfakidis, and J. Cosmas, "The MoMuSys MPEG-4 mobile multimedia terminal," in *ACTS Mobile Summit Conference*, Jun. 1998, pp. 224–229.

Junu Kim received B.S. and M.S. degrees in computer science and engineering from Seoul National University, Korea, in 1995 and 2001. Since March 2001, he has been working toward a Ph.D. in the School of Computer Science and Engineering, Seoul National University. His current research interests are in the areas of multimedia communications and quality of service in wireless networks.

Jinsung Cho received B.S., M.S, and Ph.D. degrees in computer engineering from Seoul National University, Korea, in 1992, 1994 and 2000. He was a visiting researcher at IBM T.J. Watson Research Center in 1998, and a research staff member at Samsung Electronics from 1999 to 2003. Currently, he is an assistant professor in the School of Electronics and Information at Kyung Hee University, Youngin, Korea. His research interests include mobile networking and computing, embedded systems and software.

Heonshik Shin received a B.S. in applied physics from Seoul National University, Korea, and a Ph.D. in computer engineering from the University of Texas at Austin. He is currently a professor of Computer Science and Engineering at Seoul National University. His research interests include real-time embedded system, mobile computing, and multimedia computing.