EURASIP Journal on
Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH **Open Access**

Optimal modulation and coding scheme allocation of scalable video multicast over IEEE 802.16e networks

Chia-Tai Tsai, Rong-Hong Jan[*] and Chien Chen

Abstract

With the rapid development of wireless communication technology and the rapid increase in demand for network bandwidth, IEEE 802.16e is an emerging network technique that has been deployed in many metropolises. In addition to the features of high data rate and large coverage, it also enables scalable video multicasting, which is a potentially promising application, over an IEEE 802.16e network. How to optimally assign the modulation and coding scheme (MCS) of the scalable video stream for the mobile subscriber stations to improve spectral efficiency and maximize utility is a crucial task. We formulate this MCS assignment problem as an optimization problem, called the total utility maximization problem (TUMP). This article transforms the TUMP into a precedence constraint knapsack problem, which is a NP-complete problem. Then, a branch and bound method, which is based on two dominance rules and a lower bound, is presented to solve the TUMP. The simulation results show that the proposed branch and bound method can find the optimal solution efficiently.

Keywords: Adaptive modulation and coding, Branch and bound algorithm, IEEE 802.16e, Resource allocation, Scalable video coding

1 Introduction

With the popularity of wireless networks, the need for network bandwidth is growing rapidly. In order to provide high quality service, various categories of broadband wireless network techniques, e.g., IEEE 802.16e (or WiMAX, Worldwide Interoperability for Microwave Access) and 3GPP LTE, have been proposed. Among these techniques, IEEE 802.16e is an emerging network technique and has been deployed in many metropolises (e.g., Chicago, Las Vegas, Seattle, Taipei and so forth [1,2]). It provides mobile users with a high data rate (up to 75 Mbps) and a large coverage range (up to a radius of 10 miles) [3-5]. In addition, it also enables new classes of real-time video services, such as IPTV services, video streaming services, and live TV telecasts, which require a large transmission bandwidth, and need identical content to be delivered to several mobile stations. The most efficient way to provide such services is to use wireless multicasting, sending one copy of the video stream to multiple subscriber stations via a shared multicast channel, instead of sending multiple copies via several dedicated channels [6]. In this way, wireless multicasting can reduce bandwidth consumption significantly.

IEEE 802.16e supports a variety of modulation and coding schemes (MCSs), such as QPSK, 16QAM, and 64QAM, and allows these schemes to change on a burst-by-burst basis per link, depending on channel conditions [3-5]. Adaptive modulation and coding (AMC) is a term used in wireless communications to denote the matching of the modulation and coding to the channel condition for each subscriber station. It is widely applied to wireless networks. For example, the IEEE 802.16e base station (BS) can assign an appropriate MCS to each mobile subscriber station (MSS) based on its channel quality. This can be done by having the MSS advise its downlink channel quality indicator to the BS. The BS scheduler can take into account the channel quality of the MSS and assign an appropriate MCS for each of them so that the throughput is maximized.

Owing to the mobility (i.e., the ability to move within the coverage area) of the MSS, the signal-to-noise ratio

* Correspondence: rhjan@cs.nctu.edu.tw
Department of Computer Science National Chiao Tung University 1001 University Road, Hsinchu 300, Taiwan

(SNR) from the BS may become degraded (i.e., the MSS could be in poor channel condition at some time). The adaptation strategy for the MSS with the worst channel condition will cause the data rate to be low, especially when the multicast group size is large [7]. For example, as shown in Figure 1, the BS chooses QPSK, the most conservative and robust MCS, to accommodate all MSSs in the multicast group, even if there are some MSSs (e.g., MSS1, MSS2, and MSS3) that can be accommodated with a higher data rate MCS (e.g., 16QAM). That is, the multicast data rate is determined by the MSS which has the worst channel condition (e.g. MSS 4). As a result, the spectral efficiency tends to be poor.

The scalable video coding (SVC) scheme [8] allows for the delivery of a decodable and presentable quality of the video depending on the MSS' channel quality. The SVC scheme divides a video stream into one base layer and several enhancement layers [8]. The base layer provides a basic video quality, frame rate, and resolution of the video, and the enhancement layers can refine the video quality, frame rate, and resolution. Figure 2 shows the video quality under various combinations of video layers. The more video layers an MSS receives, the better video quality it can get. In this article, we apply the utility [9,10] to measure the satisfaction degree of the video quality that the MSS received.

In wireless networks, because the air resources are limited and shared by all receivers, organizing the layering structure of a video stream and assigning the appropriate MCS for each video layer to maximize the total utility is a crucial task [11-21]. Formally, the problem can be stated as follows: consider a video multicasting network having a scalable video stream V consisting of m video layers $L = \{l_1, l_2, ..., l_m\}$ and adaptive MCS consisting of n MCSs $\{M_1, M_2, ..., M_n\}$. The BS chooses a layering structure (i.e., selecting a set of video layers L' from L), which will multicast to the MSSs, and determines an appropriate MCS for each video layer in L' such that the total utility is the maximized subject to a bandwidth constraint.

In this article, we formulate the MCS assignment of the layering structure as a total utility maximization problem (TUMP). This article transforms the TUMP into a precedence constraint knapsack problem, which is a NP-complete problem [22]. The precedence-constraint knapsack problem is a generalization of the knapsack problem, which includes the constraint on the packed order of the items. For example, if item i precedes item j, then item j can only be packed into the knapsack if item i is already packed into the knapsack. Because the solution space of the problem TUMP consists of a large number of fruitless candidates, a branch and bound method which is based on two dominance rules and a lower bound is presented to solve the TUMP. The simulation results show that the proposed branch and bound method can find the optimal solution efficiently. Because the optimal solution can be found with just a little computation time, the proposed method is suitable for MCS assignment in a scalable video multicast over IEEE 802.16e networks.

This article is organized as follows: In Section 2, we describe and formulate the TUMP problem. We

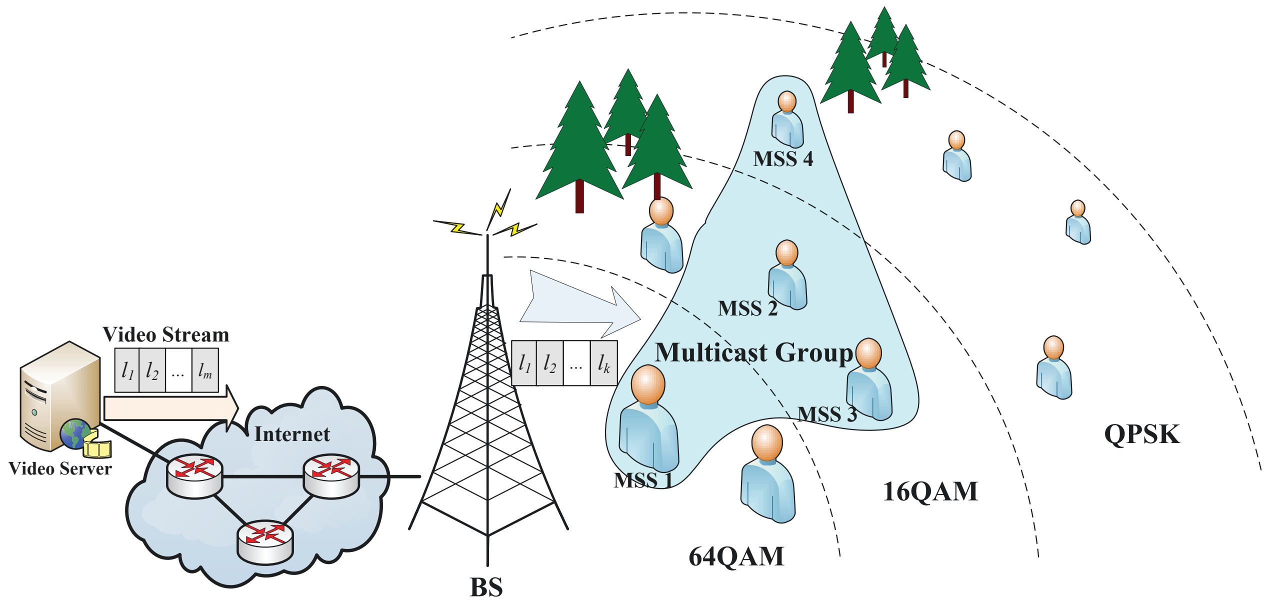

Figure 1 The video multicast network environment over IEEE 802.16e networks.

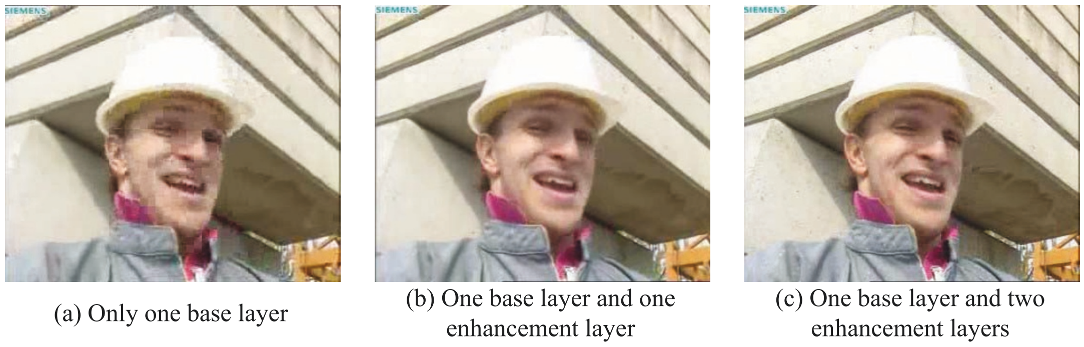
(a) Only one base layer
(b) One base layer and one enhancement layer
(c) One base layer and two enhancement layers

Figure 2 The video quality for the MSS under various numbers of video layers (the video, foreman, is downloaded from the video trace library [27]). (a) Only one base layer. (b) One base layer and one enhancement layer. (c) One base layer and two enhancement layers.

transform the TUMP into a precedence constraint knapsack problem and propose a branch and bound method to solve the TUMP in Section 3. The experimental results are given in Section 4. Finally, we conclude this article in Section 5.

2. Problem description

2.1. Statement of the problem

In this article, we consider a video multicast network environment over an IEEE 802.16e network as shown in Figure 1. The MSSs can access the Internet through the BS. The ranging process occurs when an MSS joins the network and updates periodically; hence, the BS can obtain the link quality of each MSS [3-5]. Suppose that there is a set of MSSs joined to a multicast group and subscribing to a scalable video stream V consisting of m video layers $L = \{l_1, l_2,..., l_m\}$. The video server delivers V to the BS through the Internet. The BS has n MCSs $\{M_1, M_2,..., M_n\}$. It takes each MSS's channel quality and the number of available time slots into account before organizing the layering structure. If the number of available time slots is not large enough, then the BS has to choose a set of feasible video layers L' from L and determine an appropriate MCS for each video layer in L'. Our goal is to maximize the total utility under a bandwidth constraint.

2.2 Model and notations

Based on the specification of IEEE 802.16e [3-5], each frame consists of subchannels and OFDMA symbols. For the down link frame, a time slot, the minimum allocable resource unit, includes two consecutive OFDMA symbols in a subchannel [3-5]. Let S be the number of the available time slots allocated to the video stream. The MCSs, $M_1, M_2,... M_n$, are sorted in ascending order from the lowest data rate (i.e., the most robust) MCS to the highest data rate MCS. Let r_j be the data rate (bytes per time slot) of M_j, $j = 1, 2,..., n$, and $r_1 \leq r_2 \leq ... \leq r_n$. For example, as shown in Figure 1, the BS supports three MCSs QPSK, 16QAM, and 64QAM, i.e., M_1 = QPSK, M_2 = 16QAM, and M_3 = 64QAM.

Suppose that the MSS receives a set of video layers $L' = \{l_1, l_2,..., l_k, l_x, l_y,..., l_z\}$ from a BS where $k + 1 < x < y < z$. It is noted that an enhancement layer, say layer l_k, can be used to refine the video quality only when the MSS has received all the lower layers, i.e., $l_1, l_2,..., l_{k-1}$ [13]. Therefore, in this example, the maximum number of consecutive video layers of L' is k. Then, we say that the received enhancement layers $l_2, l_3,..., l_k$ are the *valid* video layers for refining the video quality. The invalid video layer (e.g., l_x, l_y, or l_z) will be discarded by the MSS.

In order to determine the satisfaction degree of the video quality for an MSS, a relative measure of satisfaction, called *utility*, is used in [11-21]. Figure 3 is an example of the utility function for MSS under various numbers of video layers [10]. When an additional video layer is received, the utility is increased and the MSS can experience the additional satisfaction. Because the attenuation is caused by shadowing or slow fading in

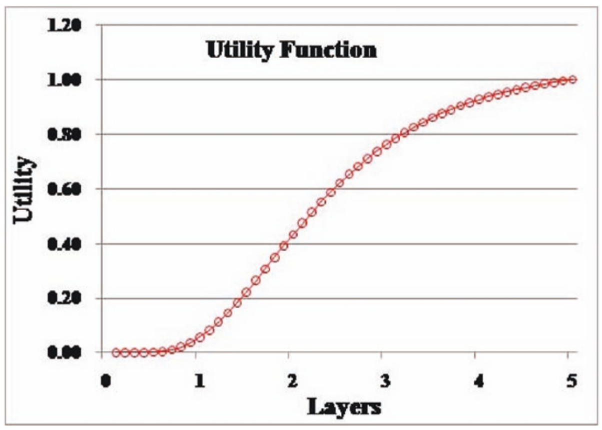

Figure 3 Utility function under various numbers of video layers.

the wireless communication, the utility function is often assumed to be log-normally distributed [23]. Let $Util(i)$ be the utility of MSS when it has received i valid video layers. Let δ_i be the additional utility when the MSS received the i^{th} video layer, i = 1, 2,..., m. Then, δ_i can be calculated as follows:

$$\delta_i = Util(i) - Util(i-1) \tag{1}$$

It is noted that $Util(0) = 0$. Thus, the additional utility of the base layer, δ_1, equals $Util(1)$. Table 1 lists the utility and additional utility of the MSS under various numbers of video layers (e.g., $m = 5$).

Let u_j be the number of MSSs which can receive the video stream encoded by M_j. The number of MSSs at lower MCSs (e.g., QPSK) is greater than that at higher MCSs (e.g. 64QAM), i.e., $u_1 \geq u_2 \geq ... \geq u_j$. For example, Table 2 lists the set of MCSs which can be accepted by the MSSs in the multicast group as shown in Figure 1. From Table 2 we can find $u_1 = 4$, $u_2 = 3$, and $u_3 = 1$.

Let w_{ij} be the amount of utility when the video layer l_i is encoded by M_j. We can compute w_{ij} by $w_{ij} = \delta_i u_j$, i = 1, 2,..., m and j = 1, 2,..., n. It is noted that $w_{i1} \geq w_{i2} \geq ... \geq w_{ij}$, i = 1, 2,..., m because $u_1 \geq u_2 \geq ... \geq u_j$. In addition, suppose that the video layer l_i contains λ_i bytes, i = 1, 2,..., m. The number of time slots t_{ij} required to transmit the layer l_i using MCS M_j can be computed by

$$t_{ij} = \left\lceil \frac{\lambda_i}{r_j} \right\rceil, \quad \text{where } i = 1, 2, \ldots, m \text{ and } j = 1, 2, \ldots, n. \tag{2}$$

2.3 Problem formulation

The optimal MCS assignment for scalable video multicast can be mathematically stated as follows.

Problem TUMP:

$$\text{Maximize } z = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} x_{ij} \tag{3}$$

$$\text{Subject to } \sum_{i=1}^{m} \sum_{j=1}^{n} t_{ij} x_{ij} \leqslant S \tag{4}$$

$$\sum_{j=1}^{n} x_{ij} \leq 1, \ i = 1, 2, \ldots, m \tag{5}$$

Table 1 Utility and additional utility of an MSS under various numbers of video layers

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
$Util(i)$	0.06	0.43	0.76	0.93	1
δ_i	0.06	0.37	0.33	0.17	0.07

Table 2 The set of MCSs which can be accepted by the MSSs in the multicast group

	The set of MCSs that can be received by the MSS
MSS1	$\{M_1, M_2, M_3\}$
MSS2	$\{M_1, M_2\}$
MSS3	$\{M_1, M_2\}$
MSS4	$\{M_1\}$

$$\sum_{j=1}^{n} x_{i-1j} - \sum_{j=1}^{n} x_{ij} \geq 0, \quad i = 2, 3, \ldots, m \tag{6}$$

$$x_{ij} = 0 \text{ or } 1, \ i = 1, 2, \ldots, m \text{ and } j = 1, 2, \ldots, n \tag{7}$$

This is a 0-1 integer programming problem. x_{ij} is the decision variable where $x_{ij} = 1$ indicates that video layer l_i is encoded by M_j; otherwise, $x_{ij} = 0$. Constraint (4) ensures that the sum of the required time slots cannot exceed S. Constraint (5) limits a video layer to being encoded by only one MCS at the same time. In order to avoid sending the invalid video layer, constraint (6) ensures that the video layer l_i can only be encoded if the video layer l_{i-1}has been encoded.

3. The solution method

In this section, we first transform the TUMP into a precedence constraint knapsack problem, which is a well-known NP-complete problem [22]. Then, we propose a branch and bound algorithm for solving the TUMP problem.

3.1. Problem hardness

We convert the inequality constraint of the TUMP problem (Equation 5) to the equality constraint by introducing a set of slack variables X, where $X = \{x_{1n+1}, x_{2n+1}, ..., x_{mn+1}\}$. For all i, $x_{i\ n+1}$is defined as

$$x_{in+1} = \begin{cases} 0, \text{ if the video } l_i \text{ is encoded by } M_j \\ 1, \text{ otherwise} \end{cases} \tag{8}$$

That is, $x_{in+1} = 1 - \sum_{j=1}^{n} x_{ij}$, where i = 1, 2,..., m. For all i, let w_{in+1}= 0 and t_{in+1}= 0. We can rewrite Equations 3, 4, and 5 as follows:

$$z = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} x_{ij} = \sum_{i=1}^{m} (w_{i1} x_{i1} + w_{i2} x_{i2} \cdots + w_{in+1} x_{in+1}) = \sum_{i=1}^{m} \sum_{j=1}^{n+1} w_{ij} x_{ij} \tag{9}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} t_{ij} x_{ij} = \sum_{i=1}^{m} (t_{i1} x_{i1} + t_{i2} x_{i2} \cdots + t_{in+1} x_{in+1}) = \sum_{i=1}^{m} \sum_{j=1}^{n+1} t_{ij} x_{ij} \leq S \tag{10}$$

$$\sum_{j=1}^{n} x_{ij} + x_{in+1} = \sum_{j=1}^{n+1} x_{ij} = 1, i = 1, 2, \ldots, m. \tag{11}$$

From Equation 6, we know that $\sum_{j=1}^{n} x_{i-1j} \geq \sum_{j=1}^{n} x_{ij}$. It is noted that $\sum_{j=1}^{n} x_{i-1j} + x_{i-1n+1} = \sum_{j=1}^{n} x_{ij} + x_{in+1} = 1$. Thus, Equation 6 can be transformed as follows:

$$x_{i-1n+1} \leqslant x_{in+1}, i = 2, 3, \ldots, m. \quad (12)$$

Therefore, the TUMP problem can be transformed as follows:

Problem TUMP1:

$$\text{Maximize}\, z = \sum_{i=1}^{m} \sum_{j=1}^{n+1} w_{ij} x_{ij} \quad (13)$$

$$\text{Subject to} \sum_{i=1}^{m} \sum_{j=1}^{n+1} t_{ij} x_{ij} \leq S \quad (14)$$

$$\sum_{j=1}^{n+1} x_{ij} = 1, \text{ where } i = 1, 2, \ldots, m \quad (15)$$

$$x_{1n+1} \leq x_{2n+1} \leq \cdots \leq x_{mn+1} \quad (16)$$

$$x_{ij} = 0 \text{ or } 1, i = 1, 2, \ldots, m \text{ and } j = 1, 2, \ldots, n+1 \quad (17)$$

It is noted that the above problem TUMP1 is equivalent to the precedence constraint knapsack problem [22], which is a NP-complete problem.

3.2. Branch and bound algorithm

In this section, we propose a branch and bound algorithm, which is commonly employed to solve integer programming problems [24,25], for solving the TUMP problem. Obviously, the solution space of TUMP may consist of all 2^{mn} combinations of the mn binary variables. However, we can apply the multiple choice constraints (5) and the precedence constraints (6) to reduce the solution space to $\binom{m+n}{n}$ combinations. Figure 4 shows a possible tree organization for the case $m = 4$ and $n = 3$. We call such a tree a combinatorial tree. The links are labeled by possible choices of M_j for l_i (i.e., $x_{ij} = 1$). For example, links from the root (level-0) node to level-1 nodes specify that each of x_{1j}, $j = 1, 2, \ldots, n$, is selected and set to 1. The links from the level-i node, pointed to by the link with label $x_{ij} = 1$, to level-(i + 1) nodes are labeled by $x_{i+1j} = 1$, $x_{i+1j+1} = 1, \ldots$, or $x_{i+1n} = 1$ due to the precedence constraints. For example, there are only two links from node 13 at level-2, pointed to by the link with label $x_{22} = 1$, to the level-3 nodes 14 and 17. They are labeled $x_{32} = 1$ and $x_{33} = 1$, respectively. Thus, the solution space is defined by all paths from the root node to any node in the tree. The possible paths are () (this corresponds to the empty path from the root to itself); ($x_{11} = 1$); ($x_{11} = 1$, $x_{21} = 1$); ($x_{11} = 1$, $x_{21} = 1$, $x_{31} = 1$); ($x_{11} = 1$, $x_{21} = 1$, $x_{31} = 1$, $x_{41} = 1$); ($x_{11} = 1$, $x_{21} = 1$, $x_{31} = 1$, $x_{42} = 1$); ($x_{11} = 1$, $x_{21} = 1$, $x_{31} = 1$, $x_{43} = 1$); ($x_{11} = 1$, $x_{21} = 1$, $x_{32} = 1$); ($x_{11} = 1$, $x_{21} = 1$, $x_{32} = 1$, $x_{42} = 1$); etc. The path ($x_{1\gamma_1} = 1, x_{2\gamma_2} = 1, \ldots, x_{i\gamma_i} = 1$) defines a possible solution that $x_{1\gamma_1} = 1, x_{2\gamma_2} = 1, \ldots, x_{i\gamma_i} = 1$ and the others x_{ij} equals zero. There are $\binom{m+n}{n} = \binom{3+4}{4} = 35$ nodes in Figure 4. That is, there are 35 possible combinations for selecting M_j, $j = 1, 2, 3$ for l_i, $i = 1, 2, 3, 4$.

To find an optimal solution, we do not consider all combinations, since it is time-consuming. We apply the greatest utility branch and bound algorithm to find the optimal solution by traversing only a small portion of the combinatorial tree. The branch and bound method has three decision rules that provide the method for:

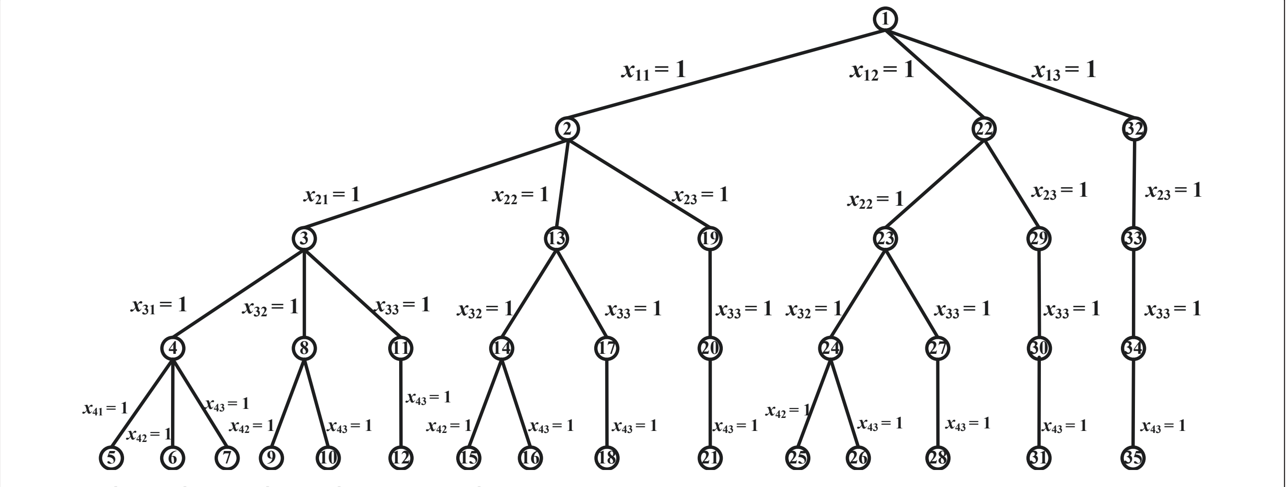

Figure 4 The combinatorial tree where m = 4 and n = 3.

1. Estimation of the upper bound of the objective function (i.e., total utility) at every node of the combinatorial tree.
2. Feasibility test at each node.
3. Selecting the next live node for branching and terminating the algorithm.

3.2.1 Estimation of the upper bound of the objective function at each node

Let p be the current node in the combinatorial tree and $(x_{1y_1} = 1, x_{2y_2} = 1, \ldots, x_{iy_i} = 1)$ be the path from the root to the node p. Let $f(p)$ be the total utility received at node p (i.e., $f(p) = w_{1y_1} + w_{2y_2} + \cdots + w_{iy_i}$). Let $g(p)$ be the maximum total utility that appears in the solutions generated from node p.

$$g(p) = f(p) + \sum_{k=i+1}^{m} w_{kyi} \tag{18}$$

Equation 18 results from $w_{ky_i} \geq w_{ky_{i+1}} \geq \cdots \geq w_{ky_m}, k = i+1, i+2, \ldots, m$.

3.2.2 Feasibility test at each node

Whenever a node is visited, the feasibility test, asking for the required number of time slots which cannot exceed S (see constraint (4)), is applied. Let p be the visiting node in the tree and $(x_{1y_1} = 1, x_{2y_2} = 1, \ldots, x_{iy_i} = 1)$ be the path from the root to the node p. Thus, the total number of time slots consumed so far can be computed by $h(p) = t_{1y_1} + t_{2y_2} + \cdots + t_{iy_i}$. If $h(p) \leq s$, node p is feasible; otherwise, node p is infeasible.

3.2.3 Selection of a branching node and termination condition

To handle the generation of the combinatorial tree, a data structure (live-node list) records all live nodes that are waiting to be branched. Initially, the child nodes of the root node are generated and added to the live-node list. The search strategy of the branch and bound algorithm is the greatest utility first. That is, the node, say p, selected for next branching is the live node the $g(p)$ of which is the greatest among all the nodes in the live-node list. If node p is feasible, then the child nodes of p are added to the live-node list. For example, if node 3 is feasible and selected for branching, then three nodes, 4, 8, and 11 are generated and added to the live-node list (see Figure 4).

Traversal of the combinatorial tree starts at the root node and stops when the live-node list is empty. In addition, a lower bound of total utility (LT) is associated with the branch and bound algorithm. LT = 0, initially, and is updated to be max (LT, $f(u)$) whenever a feasible node u is reached. If node p satisfies $g(p) \leq$LT (i.e., the maximum total utility of node p is smaller than or equal to the lower bound total utility of the current optimal solution), then it is bounded since further branching from p does not lead to a better solution. If node p is infeasible, then it is bounded since further branching from p does not lead to a feasible solution. When any branch is terminated, the next live-node is chosen by the greatest utility policy. If the live-node list becomes empty, the optimal solution is defined by the path from the root to the node w with $f(w) = LT$. Optimal utility LT is the output of Figure 5.

Numerical example and results

4.1. A numerical example

Consider an example of a scalable video with four video layers (i.e., l_1, l_2, l_3, l_4). The BS supports three MCSs (i.e., M_1, M_2, M_3). Suppose that $[\delta_i] = [0.4\ 0.3\ 0.2\ 0.1]^{\mathrm{T}}$, $[u_j] = [7\ 3\ 2]$, and $[r_j] = [48\ 96\ 192]$ (bits per time slot). We assume that each video layer has the same size, λ = 192 bits per frame; that is, $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda$. We then assume that the number of available time slots S = 21. The number of required time slots $[t_{ij}]$ and the total utility $[w_{ij}]$ can be found as follows:

$$[t_{ij}] = \begin{bmatrix} 8 & 4 & 2 \\ 8 & 4 & 2 \\ 8 & 4 & 2 \\ 8 & 4 & 2 \end{bmatrix},$$

$$[w_{ij}] = \begin{bmatrix} 2.8 & 1.2 & 0.8 \\ 2.1 & 0.9 & 0.6 \\ 1.4 & 0.6 & 0.4 \\ 0.7 & 0.3 & 0.2 \end{bmatrix}.$$

First, as shown in Figure 6, the algorithm checks if node 1 is a feasible node or not. Because $h(1) = 0$ which is smaller than 21, node 1 is a feasible node. The current total utility is $f(1) = 0$. Then, the algorithm adds nodes 2, 22, and 32 to the live-node list and computes $g(2)$, $g(22)$, and $g(32)$. By Equation 18, we obtain:

$$\begin{aligned}
g(2) &= f(2) + w_{21} + w_{31} + w_{41} = w_{11} + w_{21} + w_{31} + w_{41} = 2.8 + 2.1 + 1.4 + 0.7 = 7,\\
g(22) &= f(22) + w_{22} + w_{32} + w_{42} = w_{12} + w_{22} + w_{32} + w_{42} = 1.2 + 0.9 + 0.6 + 0.3 = 3,\\
g(32) &= f(32) + w_{23} + w_{33} + w_{43} = w_{13} + w_{23} + w_{31} + w_{41} = 0.8 + 0.6 + 0.4 + 0.2 = 2.
\end{aligned}$$

Since $g(2) = 7$ is the greatest value among nodes 2, 22, and 32, the algorithm chooses node 2 for branching(see Figure 6).

Next, the algorithm checks the feasibility of node 2 (see Figure 7). Because $h(2) = t_{11} = 8 < 21$, node 2 is a feasible node. The current total utility is $f(2) = w_{11}$ = 2.8. Then, the algorithm adds nodes 3, 13, and 19 to the live-node list. Because $g(3) = f(3) + w_{31} + w_{41} = (w_{11} + w_{21}) + w_{31} + w_{41} = (2.8 + 2.1) + 1.4 + 0.7 = 7$ is the greatest value among nodes 3, 13, and 19, it chooses node 3 for branching.

```
1  Initialize the live-node list to be empty;
2  Put root node $v_1$ on the live-node list;
3  Set $f(v_1) := 0$;
4  Set $LT := 0$;
5  while live-node list is not empty do
6  begin
7      choose node $p$ with the greatest value of $g(p)$ from the live-node list;
8      Set $G := 0$;
9      if $h(p) > S$ then
10         remove node $p$ from the live-node list;
11     else begin
12         Put the child nodes of node $p$ into set $G$;
13         for each node $u$ in $G$ do
14             begin
15                 if $g(u) > LT$ then
16                     set max $(LT, f(u))$;
17             end;
18             insert node $u$ into the live-node list;
19         end;
20     remove node $p$ from the live-node list;
21     end;
22 end;
23 output the answer: node $w$ and the optimal value $g(w) := LT$;
```

Figure 5 Branch and bound algorithm for solving the problem TUMP.

Because $h(4) = t_{11} + t_{21} + t_{31} = 24 > 21$, node 4 is infeasible and gets killed (or bounded). By the same method, the algorithm chooses node 8 for branching (see Figure 8). Since $h(9) = 24 > 21$ and $h(10) = 22 > 21$, nodes 9 and 10 get killed. The algorithm finds the next node for branching from the live-node list. Since $g(p)$, $p = 11, 13, 19, 22, 23$, which are smaller than or equal to LT = $f(8) = 5.5$, nodes 11, 13, 19, 22, and 32 are bounded. Now, the live-node list is empty and then the algorithm will be terminated. The maximum utility answer node is node 8. It has a utility of 5.5. That is, the optimal solution is ($x_{11} = 1$, $x_{21} = 1$, $x_{32} = 1$). The video layers l_1, l_2, and l_3 are selected to be delivered and are encoded by M_1, M_1, and M_2, respectively.

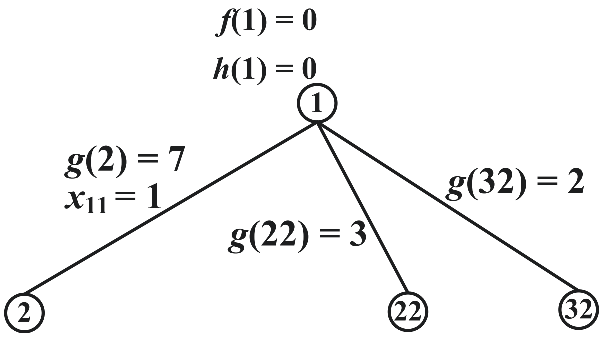

Figure 6 The algorithm chooses node 2 for branching.

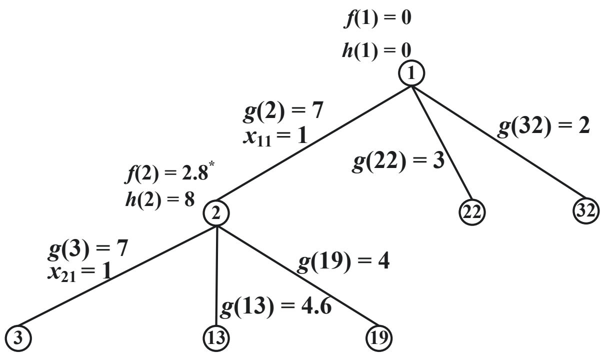

Figure 7 The algorithm chooses node 3 for branching and the current optimal solution is ($x_{11} = 1$) and current total utility is $z^* = f(2) = 2.8$.

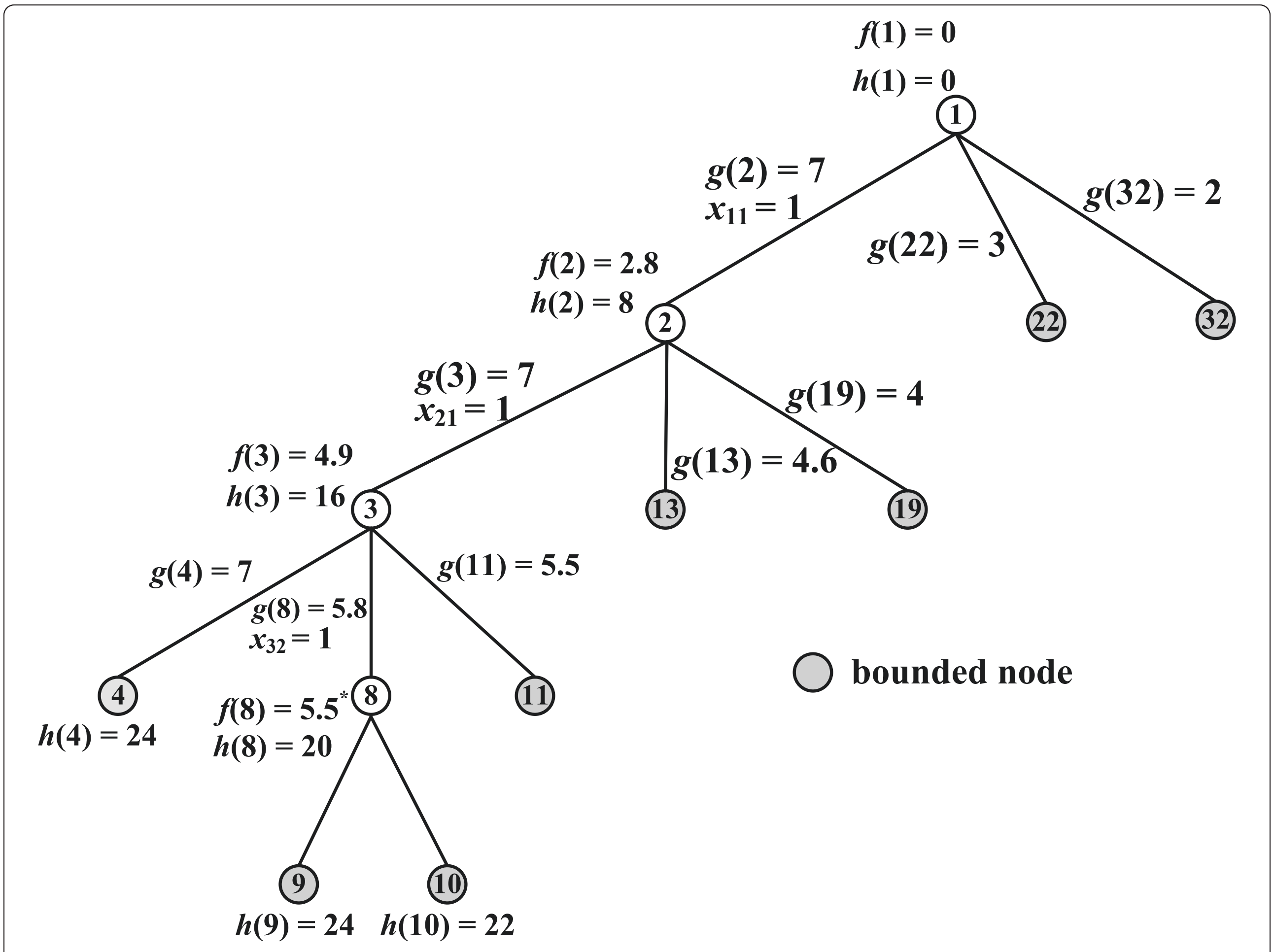

Figure 8 The optimal solution of a video stream is (x_{11} = 1, x_{21} = 1, x_{32} = 1) and optimal total utility z^* = $f(8)$ = 5.5.

4.2. Experimental results

We have conducted simulations to demonstrate how effective the proposed mathematical model is. The simulation ran on a BS with 100 MSSs which were randomly placed within a cell. The coverage area of the BS was divided into six rings, P_1, P_2,..., and P_6 as shown in Figure 9. Six types of MCS as in the IEEE 802.16e standard [3-5] were used (i.e., n = 6). The MSS in rings P_1, P_2,..., and P_6 can be accommodated with MCS sets $\{M_1, M_2, M_3, M_4, M_5, M_6\}$, $\{M_1, M_2, M_3, M_4, M_5\}$,..., and $\{M_1\}$, respectively. The video stream was divided into one base layer and six enhancement layers (i.e., m = 7). The utility function was assumed to be log-normally distributed due to the attenuation caused by shadowing or slow fading in the wireless communication. The shape parameter and the scale parameter of the utility function were set to 1.5 and 0.5, respectively (see Figure 3) [10].

Three assigning MCS methods were considered in the simulation:

1). The naive method: It chooses the highest MCS, which can be received by all MSSs in the multicast group, to encode the video layers, and allocates the available timeslots to the video layers one by one until the remaining timeslots cannot accommodate the next layer.

2). The uniform method [26]: It chooses the highest MCS, which can be received by all MSSs in the multicast group, to encode the base layer. Next, the uniform algorithm chooses the MCS which covers at least 60% of the MSSs in the multicast group to encode the enhancement layers.

3). The proposed method: It solves the TUMP problem to find the optimal MCS for each video layer by the branch and bound algorithm.

The total utility values achieved by the naive method, the uniform method, and the proposed method are denoted by X_{naive}, X_{uni}, and X_{opt}, respectively. The

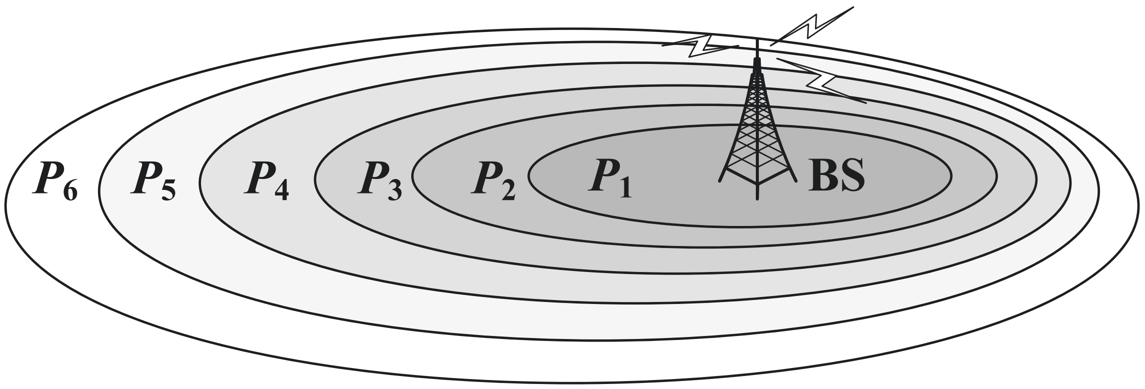

Figure 9 **The coverage area of the BS with six rings**.

comparisons among X_{naive}, X_{uni}, and X_{opt} are made (shown in Figure 10). Each data point in Figure 10 is the average over 10 runs. The results show that the total utility values X_{opt} are greater than X_{uni} or X_{naive}. The gaps among X_{naive}, X_{uni}, and X_{opt} are larger when the available bandwidth is in the range of 1500-3000 timeslots/s.

Figure 11 shows one sample of the simulation results for the optimal algorithm and the uniform algorithm with the number of available timeslots S = 2500. As shown in Figure 11a, for both algorithms, the MSS can receive more video layers when it is more close to the BS. However, the numbers of video layers delivered by the optimal algorithm to the MSS in all rings except ring P_6 are greater than or equal to the numbers of video layers delivered by the uniform algorithm. Similarly, from Figure 11b, it is noted that the utility values achieved by the optimal algorithm are greater than or equal to the values achieved by the uniform algorithm for all rings except ring P_6. In this sample, the numbers of the MSSs for rings P_1, P_2, P_3, P_4, P_5, and P_6 were 3, 5, 42, 7, 10, and 33, respectively. The total utility achieved by the proposed algorithm was 40.92 (= (3 + 5 + 42) × 0.71 + 7 × 0.47 + 10 × 0.18 + 33 × 0.01), while that achieved by the uniform algorithm was 34.53 (= (3 + 5 + 42 + 7) × 0.47 + (10 + 33) × 0.18). The optimal algorithm shows its benefit.

On the other hand, we also present the computational experiments to show the effectiveness of the branch and

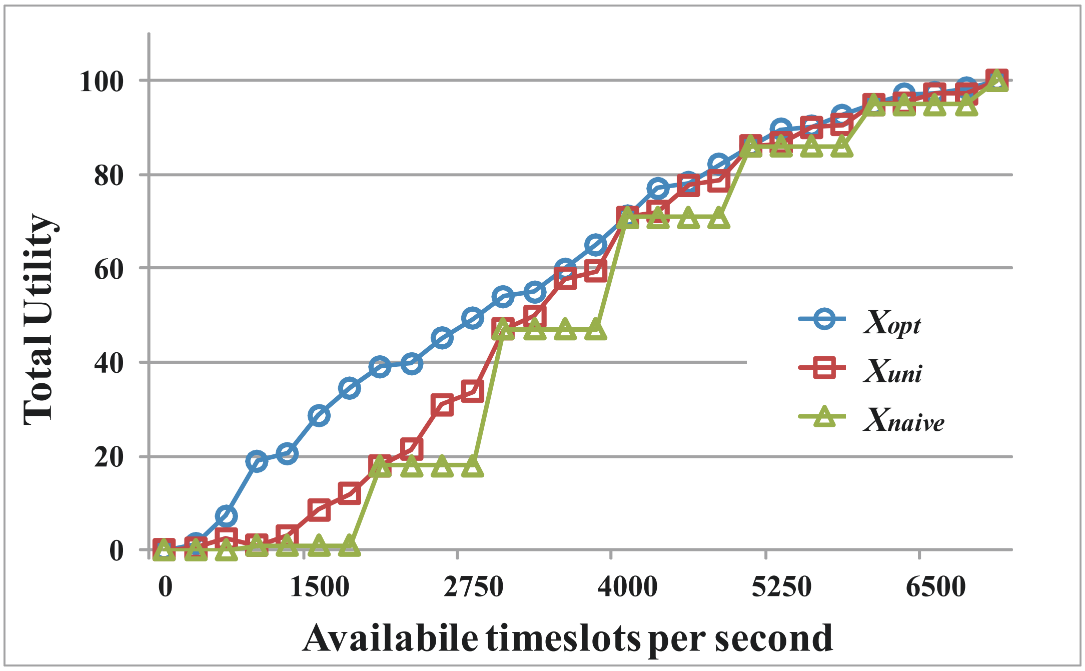

Figure 10 **The utility of the optimal solution, the uniform algorithm, and the naive algorithm with different available timeslots per second**.

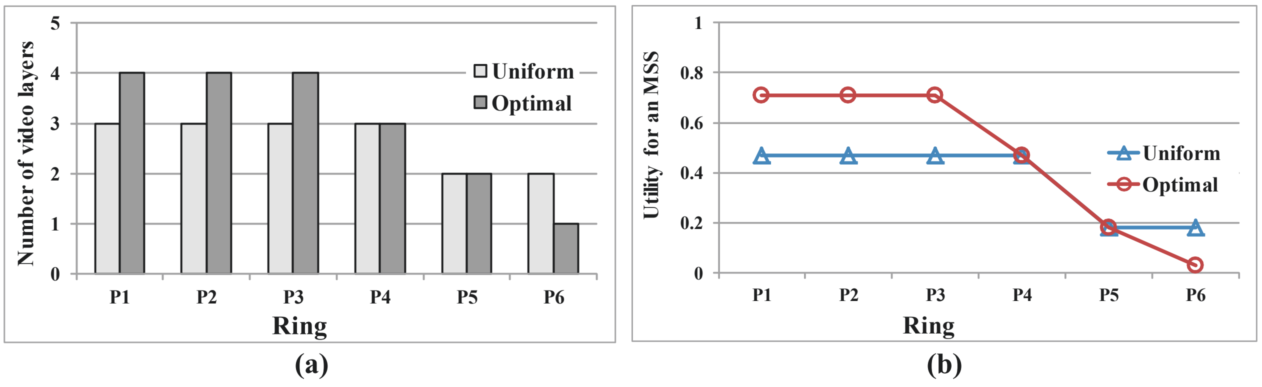

Figure 11 The number of video layers that an MSS can receive and the utility of an MSS under various rings when the available timeslots (S) equal to 2500.

bound algorithm. The real execution times of the algorithm depend on the number of video layers (m), the number of MCSs (n), and the number of available time slots (S). The experiments were conducted on a desktop PC with an Intel Core 2 Duo 1.6GHz processor and 2 GB memories. The operating system was Windows XP. The programs were coded in C and are available from the corresponding author upon request.

The simulation also ran on a BS with 100 MSSs which were randomly placed. We assume the frame duration is 5 ms. Each MSS subscribes a scalable video, in which the video rate is 320 kbps (i.e., 1.6 kb per frame). The video rate is a measure of the rate of information content in a video stream. The video is divided equally across the number of video layers. The simulation results are summarized in Table 3 which includes the number of nodes generated, the number of computations of $f(p)$, and the execution time (CPU time). Table 3 shows that Figure 5 appreciably reduces the number of nodes generated and the number of unnecessary tries for infeasible nodes. For

Table 3 The simulation results under various numbers of MCSs, video layers, and available time slots

m	*n* = 3				*n* = 6			
	S	Computations of *f*(*p*)	Nodes generated	CPU time (μs)	*S*	Computations of *f*(*p*)	Nodes generated	CPU time (μs)
2	2×10^3	3	6	0.842	2×10^3	1	4	0.914
	4×10^3	1	3	0.634	4×10^3	1	6	1.138
	6×10^3	2	5	0.756	6×10^3	2	11	1.442
	8×10^3	2	6	0.817	8×10^3	2	12	1.618
4	2×10^3	9	9	2.928	2×10^3	6	14	4.190
	4×10^3	3	7	1.727	4×10^3	3	15	3.038
	6×10^3	4	11	2.021	6×10^3	4	22	3.500
	8×10^3	4	12	2.105	8×10^3	4	24	3.580
6	2×10^3	19	19	6.655	2×10^3	22	43	14.027
	4×10^3	4	10	2.813	4×10^3	5	22	5.220
	6×10^3	5	15	3.583	6×10^3	5	30	6.286
	8×10^3	6	18	3.831	8×10^3	6	36	6.632
8	2×10^3	22	25	9.673	2×10^3	47	96	32.430
	4×10^3	7	15	4.955	4×10^3	6	29	8.512
	6×10^3	7	21	5.583	6×10^3	7	42	9.869
	8×10^3	8	24	5.980	8×10^3	8	48	10.357
10	2×10^3	40	43	17.61	2×10^3	107	202	75.604
	4×10^3	10	20	7.397	4×10^3	13	50	16.035
	6×10^3	10	27	8.210	6×10^3	10	55	15.239
	8×10^3	10	30	8.225	8×10^3	10	60	14.747

example, if you apply an exhaustive search to a problem with size $(m, n) = (10, 6)$, then the total number of nodes is $\binom{10+6}{6} = 8008 \approx 2^{14}$. However, the number of nodes generated by Figure 5 is only $202 < 2^8$ (see Table 3) for $(m, n, S) = (10, 6, 2 \times 10^3)$ because we apply the branch and bound approach. It takes 107 tries to compute $f(p)$ for obtaining the MCS assignment of the layering structure, which is much less than 8008. This means that the dominance rules (i.e., feasibility test and utility bound) can be employed to discard the most infeasible and unnecessary nodes before computing $f(p)$. This reduces the computational time significantly.

From Table 3, the computational time to determine the optimal MCS assignment for the video layering structure is less than 75.604 μs. The computational time is small enough. Thus, the branch and bound method is effective and suitable for BS to determine the video layering structure and MCS assignment for IEEE 802.16e network multicast.

5. Conclusion

In this article, we consider an optimal MCS assignment problem which improves spectral efficiency and maximizes total utility for the scalable video multicast in IEEE 802.16e networks. We propose a branch and bound algorithm to find an optimal solution for this problem. In the experiment, it was shown that the proposed method performs well compared to the uniform method and the naïve method. The computation time of the proposed branch and bound algorithm is very small. Thus, our proposed method is suitable for BS to determine the video layering structure and the MCS assignment in the IEEE 802.16e network multicast.

Because of the Doppler Effect, when MSS is moving, the MSS's velocity causes a shift in the frequency of the signal transmitted along each signal path. It causes fast fading of the received signal for MSS. Thus, the BS will encode the video layers by the more conservative and robust MCS for the moving MSS. Therefore, the video quality for MSS when it stands at a point is better than that when it moves within the same region. Looking ahead, considering the mobility of MSS for the MCS assignment problem might be an interesting future study.

Abbreviations

AMC: adaptive modulation and coding; BS: base station; LT: lower bound of total utility; MCS: modulation and coding scheme; MSS: mobile subscriber station; SNR: signal-to-noise ratio; SVC: scalable video coding; TUMP: total utility maximization problem.

Acknowledgements

This article was supported in part by the National Science Council of the ROC, under Grants NSC -97-2221-E-009-048-MY3 and NSC-97-2221-E-009-049-MY3.

Competing interests

The authors declare that they have no competing interests.

Received: 27 October 2010 Accepted: 9 July 2011 Published: 9 July 2011

References

1. 4G Coverage, Sprint, http://www.sprint.com/
2. VMAX, http://www.vmax.net.tw/
3. IEEE Computer Society and IEEE Microwave Theory and Techniques Society, *IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems*. IEEE Standard 802.16e-2005 (2005)
4. IEEE Computer Society and IEEE Microwave Theory and Techniques Society, *IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems*. IEEE standard 802.16-2004 (2004)
5. JG Andrews, A Ghosh, R Muhamed, *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*, 1st edn. (Prentice Hall, New Jersey, 2007)
6. M Hauge, Ø Kure, Multicast in 3G networks: employment of existing IP Multicast protocols in UMTS, International Workshop on Wireless Mobile Multimedia (WoWMoM), Atlanta, USA, Sept. 2002
7. N Jindal, ZQ Luo, Capacity limits of multiple antenna multicast, in *IEEE International Symposium on Information Theory (ISIT)*, (Seattle, USA, July 2006)
8. H Schwarz, D Marpe, T Wiegand, Overview of the scalable video coding extension of the H.264/AVC standard. IEEE Trans. Circuits Syst Video Technol. **17**(9), 1103–1129 (2007)
9. S Shenker, Fundamental design issues for the future internet. IEEE J Sel Areas Commun. **13**(7), 1176–1188 (1995)
10. L Shi, C Liu, B Liu, Network utility maximization for triple-play services. Comput. Commun. **31**, 2257–2269 (2008). doi:10.1016/j.comcom.2008.02.016
11. J Liu, B Li, YT Hou, I Chlamtac, Dynamic layering and bandwidth allocation for multisession video broadcasting with general utility functions, in *The 22th IEEE International Conference on Computer Communications (IEEE INFOCOM)*, San Francisco, USA, April 2003
12. WH Kuo, T Liu, W Liao, Utility-based resource allocation for layer-encoded IPTV multicast in IEEE 802.16 (WiMAX) wireless networks, in *IEEE International Conference on Communications (ICC)*, Glasgow, Scotland, June 2007
13. P Li, H Zhang, B Zhao, S Rangarajan, Scalable video multicast in multi-carrier wireless data systems, in *The 17th IEEE International Conference On Network Protocols (ICNP)*, Princeton, USA, October 2009
14. H Chi, C Lin, Y Chen, C Chen, Optimal rate allocation for scalable video multicast over WiMAX, in *IEEE International Symposium on Circuits and Systems (ISCAS)*, Seattle, USA, May 2008
15. J Shi, D Qu, G Zhu, Utility maximization of layered video multicasting for wireless systems with adaptive modulation and coding, in *IEEE International Conference on Communications (ICC)*, Istanbul, Turkey, June 2006
16. S Deb, S Jaiswal, K Nagaraj, Real-time video multicast in WiMAX networks, in *The 27th IEEE Conference on Computer Communications (IEEE INFOCOM)*, Phoenix, USA, April 2008
17. C Huang, P Wu, S Lin, J Hwang, Layered video resource allocation in mobile WiMAX using opportunistic multicasting, in *IEEE Wireless Communication and Networking Conference (WCNC)*, Budapest, Hungary, April 2009
18. CS Hwang, Y Kim, An adaptive modulation method for multicast communications of hierarchical data in wireless networks, in *IEEE international conference on communications (ICC)*, New York, USA, April 2002
19. M Shabany, K Navaie, Es Sousa1, A utility-based downlink radio resource allocation for multiservice cellular DS-CDMA networks. EURASIP J Wirel Commun Netw. **2007**(1) (2007)
20. J Kim, D Cho, Enhanced adaptive modulation and coding schemes based on multiple channel reportings for wireless multicast systems, in *IEEE Vehicular Technology Conference (VTC 2005 Fall)*, Dallas, USA, Sept. 2005
21. H Wang, HP Schwefel, TS Toftrgaard, History-based adaptive modulation for a downlink multicast channel in OFDMA systems, in *IEEE Wireless Communication and Networking Conference (WCNC)*, Las Vegas, USA, 2008
22. H Kellerer, U Pferschy, D Psdinger, *Knapsack Problem*, (Springer, Berlin, 2004)

23. W Nelson, *Applied Life Data Analysis*, (John Wiley & Sons, Toronto, 1982), pp. 32–36
24. R Breu, C Burdet, Branch and bound experiments in 0-1 programming. Math Program Stud. **2**, 1–50 (1974)
25. R Granfinkal, G Nemhauser, Integer Programming, (Wiley, London, 1972)
26. AMC Correia, JCM Silva, NMB Souto, LAC Silva, AB Boal, AB Soares, Multi-resolution broadcast/multicast systems for MBMS. IEEE Trans Broadcast. **53**(1), 224–234 (2007)
27. Video Trace Library, Arizona State University http://trace.eas.asu.edu

doi:10.1186/1687-1499-2011-33
Cite this article as: Tsai *et al.*: Optimal modulation and coding scheme allocation of scalable video multicast over IEEE 802.16e networks. *EURASIP Journal on Wireless Communications and Networking* 2011 **2011**:33.