Dynamic Resource Allocation Framework for MooD (MBMS Operation On-Demand)

Rafael Kaliski, *Member, IEEE*, Ching-Chun Chou, Hsiang-Yun Meng, and Hung-Yu Wei, *Member, IEEE*

***Abstract*—The demand for mobile video is increasing every year. To address the strain on long term evolution (LTE) networks 3GPP introduced multimedia broadcast multicast operation (MBMS). As of LTE release 12, support for MBMS operation on-demand (MooD) was also added (MooD enables dynamic resource configuration of multicast flows). While multicast algorithms assuage the demands on the network, quality-of-service performance metrics no longer are considered an accurate measure of a user's satisfaction with the network; recent multimedia studies show that quality-of-experience (QoE) is more accurate. In order to maximize the QoE of all users in a LTE MooD system, we propose two resource allocation algorithms, both of which efficiently allocate resource blocks (RBs) based on both the demand for each live video stream and the channel conditions of the users within each group. We also compare our resource allocation algorithms against four other commonly used resource allocation algorithms. Both of our algorithms achieve a higher QoE and video quality, when compared to other commonly used resource allocation algorithms. Furthermore, our algorithms demonstrate efficient resource allocation regardless of whether or not the RBs are sufficient.**

***Index Terms*—On-Demand eMBMS, QoE, MooD, LTE.**

I. Introduction and Related Works

The DEMAND for Mobile video is increasing every year. Based on Cisco estimates [1], IP video traffic will be around 80% of all IP traffic by 2019, up from 67% in 2014. Thus efficient means of distributing video will be of key importance in terms of addressing this traffic increase.

As of Release 12, 3GPP (3rd Generation Partnership Project) decided to add LTE MBMS (Multimedia Broadcast/ Multicast Service) Operation On-Demand (MooD) [2] with support for Over-the-Top (OTT) multimedia service. MooD enables on-the-fly MBMS service configuration and seamless service migration. For example, when it becomes more efficient to run a unicast service as a MBMS service, the system may activate a previously inactive MBMS session for the service. As such, future LTE MBMS services may be dynamically configured on the fly based on each user's/group's requirements and/or the system's preferences.

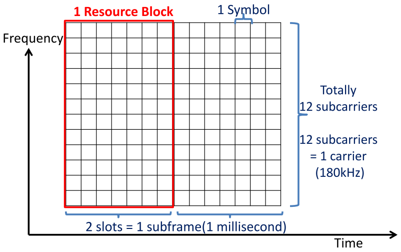


Fig. 1. LTE time-frequency radio resource.

MooD also supports dynamic MBMS configuration. To make effective use of active dynamic MBMS configuration, we examine LTE resource allocation at the Resource Block (RB) level. As shown in figure 1, RBs have a duration of 0.5 milliseconds / 1 slot with 12 subcarriers per carrier (a carrier has a width of 180kHz [3].) RBs are the basic resource allocation unit used for both unicast and multicast services. Due to the time varying wireless channel conditions experienced by end-users, several works analyze the varying channel conditions problem in terms of both multicast and broadcast services [4]. As such, we assume all resource allocation schemes exploit the fact that channel conditions can vary across RBs. In this paper we investigate multiple algorithms of RB allocation based on each users' demand and their respective per RB channel conditions. Also, in this paper the terms channels and RBs are used inter-changeably.

Prior to MooD, MBMS research such as [5]–[9] made use of Adaptive Modulation Coding (AMC) to adjust the bandwidth of the communication channel given the RB allocation (AMC depends on UE feedback of channel conditions via Channel Quality Indication (CQI) reports [3]). References [5] and [6] present several AMC strategies for multicast service defined performance metrics such as average user throughput and bit error rate in different SNR scenarios. In [7], the MBMS network was divided into concentric regions. A SVC video's layers are subsequently assigned different Modulation Coding Scheme (MCS) (less reliable, but faster, MCS are used by enhancement layers). Thus a user's location, relative to the base station, determined their SNR and which layers they could receive. This method provides a higher spectral efficiency than using the slowest MCS / MCS of the recipient with the worst SNR for all layers. Reference [8] obtains the

Manuscript received February 24, 2016; revised June 4, 2016; accepted June 16, 2016. The work of H.-Y. Wei was supported in part by CHT, in part by Foxconn, in part by Gemtek, in part by III, in part by ITRI, in part by Mediatek, in part by MoEA, in part by the Ministry of Science and Technology under Grant 103-2221-E-002-086-MY3 and Grant 102-2221-E-002-077-MY2, and in part by National Taiwan University. *(Corresponding author: Hung-Yu Wei.)*

The authors are with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan (e-mail: rkaliski@ieee.org; f95921098@ntu.edu.tw; r02921057@ntu.edu.tw; hywei@ntu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBC.2016.2590829

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

optimal MCS for multicast flows, in addition to the number of subframes to reserve for multicast, to guarantee a target bitrate for all users demanding multicast service via an exhaustive search.

To make more effective use of the instantaneously available bandwidth for a video service many works, including this one, make use of Scalable Video Coding (SVC) [10] encoded videos. An SVC video can be encoded into multiple layers based on fidelity, spatial, or temporal scalability. In this work we focus on fidelity differentiated SVC encoded video. With fidelity differentiated SVC video, a video is divided into N layers. The N layers consist of a base layer (BL) and $N - 1$ enhancement layers (EL). Each enhancement layer is dependent on the layer(s) below it. As such, should the bandwidth become insufficient to transmit all layers, a subset of the layers may be transmitted, i.e., a lower quality video can be transmitted and recovered. The work we base this work on [11], used single layer SVC encoded videos; as such, multiple video layers with potentially different sizes were never addressed.

In terms of resource allocation for SVC videos, multiple next generation related works exist. In terms of 4G systems, [9] uses a Dynamic-Programming (DP) algorithm to perform resource allocation. Per the DP algorithm result, as different channel conditions may exist, the MCS for each video layer is set. In [12], a WiMax system (WiMax uses TDMA for multiple access) with a single base station transmits multiple SVC video streams. Each video stream potentially has multiple video layers. The system attempts to maximize the system utility based on each user's channel conditions, popularity of the video program, and total available resources. Reference [13] extends this problem by addressing the issue of resource efficiency and user satisfaction in a multiple-cell system. A Hybrid Base Station scheme is proposed which dynamically determines whether a given SVC video's layers should be transmitted via multiple base stations or a single base station. For all 3 of the aforementioned works, the systems assume any given user's channel conditions are homogeneous, i.e., the system assumes there is no frequency fading across resources. As a consequence, prior methods do not work for MooD over FDD-based systems. MooD potentially supports LTE slot level RB assignment/updating the RB allocation every slot, as such the MAC-layer requires RB allocation support. Prior works only address application layer / video frame bitrates. If we only used application layer resource allocation, we would not be able to fully realize MooD's potential, i.e., resources would potentially go to waste. Furthermore, all of these systems require that the video streams be pre-encoded to derive a bitrate for each layer, i.e., none of the schemes are designed to support live video transmissions, where the video bitrate is variable. As such none of these schemes are suitable for our real-time live video streaming scenarios, which uses MAC-layer resource allocation. Our work supports multiple users with different channel conditions, does not require that we first encode the video prior to configuration, and assigns RBs for live video traffic in real-time. Furthermore, like the aforementioned 3 prior works, we also present a polynomial time algorithm to solve our optimization problem.

In terms of grouping / improving the spectral efficiency, works such as [14] attempt to optimize group formation and multimedia resource request via batching (batching attempts to reduce the amount of redundant data transmitted for flows of identical requests by delaying serving a set of requests until the batching period, as determined by the batching size and time, has elapsed.) In terms of reducing the overall bandwidth requirement, works such as [15], attempt to reduce the bandwidth requirement and buffer delay for video on demand (VOD) via periodic broadcasting of popular video segments. In terms of group formation, works such as [16] and [17], determine the optimal group assignment to improve system performance, i.e., both apply sub-group based adaptive MCS methods. Users may be re-grouped to mitigate the intrinsic inefficiencies of Conventional Multicast Schemes (CMS) related to different channel quality experienced by users. In [16], the system provides an optimal allocation of wireless resources with the goal of maximizing the proportional fair utility for both multicast and unicast users. Reference [17] performs regrouping plus optimization for multicast services. The objective is to maximize the system throughput while guaranteeing proportional fairness. Our work does not regroup users, as we are trying to maximize every user's QoE and each video stream is considered independent. We assume the amount of video data varies from one video frame to the next, i.e., requires real-time adaptation. We also assume that the channel conditions are dynamic.

Recent multimedia distribution schemes emphasize end-users' Quality of Experience (QoE) metrics over traditional Quality of Service (QoS) performance metrics, as QoE more accurately captures a users' satisfaction with a multimedia service. From a service provider's perspective, QoE metrics also provide useful insights into determining network topology / deployment of multimedia-based services.

In terms of algorithmic QoE-based works, methods such as [18] make use of Dynamic Adaptive Streaming over Hypertext Transfer Protocol (HTTP) (DASH [19]), the network conditions, utility gain of each segment, optimization in order of seconds, and a playout buffer in order to maximize the QoE of multiple users. While methods, such as [20], make use of a Pseudo Subjective Quality Assessment (PQSA) tool to perform online QoE estimation. Based on the user experienced QoE, which is estimated in real-time via a trained Random Neural Network (RNN), the video rate is dynamically adjusted. While this scheme is designed for multicast transmission, and runs at the MAC layer, it requires every multicast node runs the tool. The adaptive streaming scheme [21], Mobile-aware Adaptive Rate Control (MARC), adjusts the video transmission rate based on the channel bandwidth, i.e., the MCS is adjusted based on the clients channel conditions. Also, the SVC video quality level is adjusted based on wireless channel status (wireless channel status includes packet loss ratio, round trip time, retransmission timeout) and client buffer status. All of these aforementioned mechanisms make use of feedback mechanisms which are not supported by the LTE standard, such as client-side / receiver buffer status, QoE-reports, and channel status. In [22], the proposed algorithm ensures that all video streams transmit their respective base layer prior to allocating resources for other video streams enhancement layers. SVC video layers are selected based on their rate-quality gradient, the network conditions, and the sender buffer status, such that the video quality is maximized. The problem with their scheme is that the utility function is in terms of video quality, i.e., a rate vs quality model, an optimal solution requires training data based-off of a video stream/ pre-encoded bit stream characteristics, and the transmission buffer status / link quality is only checked every N seconds. None of the aforementioned methods are designed to instantaneously respond to changes in the channel conditions, i.e., the resource allocation is not updated at the granularity of a MAC allocation unit.

In terms of Data-driven QoE [23] related research, Dobrian *et al.* [24] utilized data-mining to analyze the relationship between video quality and user engagement. Their research found that a video's bitrate and its buffering ratio dominates a system's QoE, as captured by their user engagement metric (the user engagement metric measures the average length of time a user will watch a video before losing interest in the video and changing to a different video.) Based on Dobrian's results [24], [25] and [26] derived QoE functions for both wired and wireless multicast systems, respectively. It should be noted that [26] is also not designed to instantaneously respond to changes in the channel conditions, i.e., the resource allocation is not updated at the granularity of a MAC allocation unit.

With the advent of LTE MooD multimedia services, the MCS of each RB can be adapted based on the CQI reports from the users in the group associated with said services. To enhance the QoE of LTE MooD multimedia services we propose two SVC-aware QoE-based real-time algorithms for LTE MBMS resource allocation. In our proposed algorithms every slot, minimum MAC Allocation unit / TTI (Transmission Time Interval) of 0.5ms, the RBs are dynamically allocated to different video flows based on the system objective of reducing the buffering ratio and increasing the average bit-rate, thereby maximizing the aforementioned QoE / user engagement metric. As our algorithms do not require use of non-standard supported feedback from the network and instantaneously respond to changing channel conditions, we cannot compare our algorithms to the aforementioned QoE-based works. Our algorithms maximize the system QoE over the set of LTE MooD multimedia services provided.

In Section II we present the problem formulation. The proposed resource allocation algorithms are discussed in Section III, while a detailed description of the resource allocation algorithms is presented in Section IV. Our simulation setup, scenarios, and results are presented Section in V. We discuss fairness, efficiency, execution time, and the performance of the algorithms in Section VI. Finally, we present our conclusion and our future work in Section VII.

A. Contributions

With standard MBMS, the number of RBs is fixed for the duration of each flow. In other words, should the number of RBs need to be changed, the MBMS flow must be taken-down and setup again. Yet, with the advent of MooD, the number of RBs per MBMS flow can be modified at any given allocation period / TTI, without the need to tear-down the MBMS flow. In this work we present two new MooD MAC layer SVC-aware QoE-aware resource allocation algorithms which have the goal of maximizing the QoE of all users for live video streams; a side benefit is that the video quality is also maximized. We also compare our algorithms to several other popular resource allocation algorithms in terms of fairness/efficiency, computational complexity, and performance (The performance metrics we compare are the utility, bitrate, buffer ratio, and the number of layers delivered.)

We show both of our algorithms outperform both non-SVC aware, and non-QoE aware, algorithms, i.e., QoS-oriented algorithms cannot be directly applied to QoE problems. We also demonstrate that the computational complexity of an efficiency resource allocation algorithm need not be high. To our knowledge no other research presents a SVC-aware QoE-aware MAC layer resource allocation mechanism for live video over MooD.

II. Problem Formulation

In this section we first briefly introduce our objective function, the QoE utility, then we present the OTT live streaming resource allocation problem. Note: All notations used in this paper are summarized in table I.

A. Features of QoE Utilities

The most important parameters in the user engagement-oriented QoE utility function [24] are the buffering ratio (buffering ratio is the percentage of time spent rebuffering a video, it does not include the initial/startup buffering time) and the average bitrate. User engagement is defined as a measure of the average amount of time a user will watch a TV program/video stream before changing the TV program/starting to watch a different video stream. In terms of the aforementioned parameters, the higher the buffering ratio is, the less likely the viewers are to watch the video for an extended period of time. On the other hand, the higher the average video bitrate is, the more likely the viewers are to watch the video for a longer period of time.

The utility function in [25], and shown in equation (1), is a linear function based on the buffering ratio and the average bitrate. The exponential form of this utility function, presented in [26] and based off the same data from [24], is shown in equation (2). While simpler than its exponential counterpart, the linear utility function becomes distorted when the buffering ratio exceeds approximately 10%. The distortion is due to the limitations of linear-regression, which was used to derive the utility function. As such, the exponential utility function more accurately represents the overall user engagement curve.

$$U_{linear} = -3.7 \times \mathit{BuffRatio} + \frac{\mathit{AvgBitrate}}{20} \tag{1}$$

$$U_{exp} = \mathit{VideoLength} \times R_{\mathit{Bitrate}} \times R_{\mathit{BuffRatio}} \tag{2}$$

TABLE I
TABLE OF NOTATIONS

	Symbol	Description
Variables	$X_j^{[i]}$	Decision variable for the j^{th} channel of the i^{th} video group
	$S^{[i]}$	Trunk size for video group $[i]$ at the present moment
Input	$m_j^{[i]}$	MCS of video group $[i]$ for the j-th channel, see equation (4)
	$m_{j,k}^{[i]}$	MCS of the user-k of the video group $[i]$ for j-th channel
	$b_{F_{now},\mathcal{L}}^{[i]}$	Packet size of video layer $\mathcal{L}$ of the current video frame F_{now} in group $[i]$
	$\mathcal{L}_{max}^{[i]}$	Maximum number of layers per video group $[i]$
	$\mathcal{L}^{[i]}$	Current layer number for per video group $[i]$
	$V^{[i]}$	Set of members of the video group $[i]$
	V	Number of video groups
	$N^{[i]}$	Population of video group $[i]$, $N^{[i]} = \|V^{[i]}\|$
	N_{RB}	Number of carriers (channels) / RBs in this system
	$N_{buff}^{[i]}$	Number of buffering events for video group $[i]$
	$kbits^{[i]}$	Number of total transmitted kilobits for video group $[i]$
	$Bits^{[i]}$	Number of total transmitted bits for video group $[i]$
	T	Given allocation period, i.e. Number of Slots
	t_{now}	Time from epoch of video
	FPS	Frame Rate, Frames per second
Functions	$1 - \frac{1}{u^{[i]}}$	Probability of utility gain obtained by reducing buffering ratio
	$P[buff]$	Probability of a buffering event occurring
	$U^{[i]}$	Utility of the video group $[i]$
	U	Utility of the system (i.e. utility of this eNB)
	$M(m)$	A mapping function which maps MCS m to Bitrate per RB
	$g(n)$	A weighted function to weight a group with n users
	$RE^{[i]}$	buffering ratio effect on utility
	$DR^{[i]}$	data rate effect on utility
	$\frac{1}{u^{[i]}}$	probability of utility gain being obtained during current allocation period
	L	Allocation periods / slots to video frames scaling factor
Const	w_{re}	Coefficient of Buffering term in utility, $w_{re} = -3.7$
	w_{kb}	Coefficient of Avg.Bitrate term in utility, $w_{kb} = \frac{1}{20}$, kbps
	w_b	Coefficient of Avg.Bitrate term in utility, $w_b = \frac{w_{kb}}{1000}$, bps
ILP terms	T_{sf}	term used to scale weights to be slot relative, i.e. 0.5
	s_r	term used to scale buffering term to percentage, i.e. 100
	A_{eq}	term used to capture equality constraints
	b_{eq}	term used to capture equality constraints
	$b^{[i]}$	term used to capture group $[i]$'s inequality constraints (remaining number of bits to be transmitted, current layer)
	$A_j^{[i]}$	term used to capture inequality constraints (per RB bitrate for video group $[i]$)
	$C^{[i]}$	coefficients of objective function
	$f_j^{[i]}$	objective function given resource block j of group i
GR	f_{limit}	max incremental utility gain, limited per video layer size
	f_{max}	max incremental utility gain over all utility gains

where $R_{Bitrate}$ and $R_{BuffRatio}$ are:

$$R_{Bitrate} = 1 - e^{-0.0001024 \times AvgBitrate}$$
$$R_{BuffRatio} = e^{-0.04606 \times BuffRatio \times 100}$$

The physical meaning of equation (1) is that when the buffering ratio (buffering ratio is indicated by *BuffRatio*) increases by 1%, a corresponding decrease of user engagement by 3.7 minutes will be incurred. Also, when the average bitrate (average bitrate is indicated by AvgBitrate) increases by 20kbps, the user engagement will increase by 1 minute.

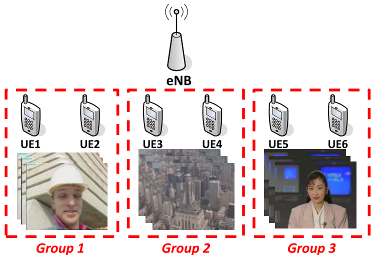


Fig. 2. Example of 3 Multicast Video Groups.

Equation (2) similarly shows that an increase in the buffering ratio or a decrease in the average bitrate results in an exponential decrease of the utility / user engagement. The additional term *VideoLength* is duration of the video, in seconds. Notice that the buffering term dominates the utility function in both functions. As demonstrated in [24], the data analysis for a live video stream (the live video stream used for the coefficients[1] is a 90-minute FIFA World cup soccer game. The coefficients -3.7 and 20 are fitting coefficients from Dobrian *et al.* [24], Figure 12 (a), and data collection bin size, respectively. Dobrian *et al.* [24] uses data-mining and curve fitting to derive these coefficients. The linear equation can also be seen in [25, p. 366]) shows that the buffering ratio is the most significant factor in determining a user's engagement, while the video bitrate is the second most significant factor in determining a user's engagement.

In the next subsection we present the test scenario.

B. Test Scenario

In this work we consider a single cell LTE eMBMS scenario. In this scenario, which is based on [27], the surrounding cells act as static interference sources with frequency flat distributions.

In our test scenario there are multiple user equipments (UEs) in the cell. Each UE subscribes to a single video stream. UEs which subscribe to the same video are classified as belonging to the same video / multicast group (In this paper the terms video and group are used interchangeably). Figure 2 shows an example of 3 separate multicast groups. Each multicast group may have multiple channels on which it can receive data. Due to the nature of multicast, all the members of a group share the same modulation coding scheme (MCS) on any given channel / carrier / RB.

There are two critical factors pertinent to resource allocation. The first factor is each video frame's packet's size and its associated deadline. The second factor is each UE's channel conditions with respect to each channel/RB. As all videos are streamed live, i.e., the eNB distributes the multimedia content on the fly. In other words, the eNB is unable to obtain information, such as a video packet's size, until the live streaming data is generated. This is a general feature of live video streaming. We assume that each UE transmits a CQI report to the eNB prior to the start of every TTI (The CQI report [3] may be configured to report the quality of all downlink carriers used

[1] The coefficients are derived from data mining of user video streams. Thus each video program / genre may have different coefficients.

by the UE. For large groups, the probability that a UE will transmit a CQI report can also be set.)

A limiting factor for transmission of the current video i with video frame F_{now} is the size of its associated video frame $b^{[i]}_{F_{now},\mathcal{L}}$, where $\mathcal{L}$ is the video layer number. The size of a video frame's packet is limited by the trunk size $S^{[i]}$, i.e., the set of RBs assigned to the video group i. Per equation (3), the resulting allocation assigns $X^{[i]}_j$ for each channel j and each group i.

$$S^{[i]} = \sum_{j=1}^{N_{RB}} X^{[i]}_j \times M(m^{[i]}_j) \times T \tag{3}$$

Where the selected modulation coding scheme (MCS) used by the j^{th} channel, assigned to video $[i]$, is $m^{[i]}_j$. $M(m^{[i]}_j)$ represents a mapping function which maps each MCS to its corresponding per RB bitrate, T represents the allocation period in terms of number of slots, and N_{RB} represents the total number of RBs assigned to the set of MBMS video services. Note: each RB may only be assigned to a single video stream at any given time.

Per equation (4), the MCS assigned to j^{th} channel is the lowest MCS among all $V^{[i]}$ of the group members in the group $[i]$. The channel conditions of the k^{th} user of video i in channel j is denoted by $m^{[i]}_{j,k}$.

$$m^{[i]}_j = \min_{k \in V^{[i]}} m^{[i]}_{j,k} \tag{4}$$

For convenience, all symbols with superscript $[i]$ indicate that they correspond to the video group $[i]$.

In order to determine the optimal trunk size for each group, we formulate the RB resource allocation problem as an optimization problem with the objective of maximizing the system utility. Prior to each TTI, the eNB runs a resource allocation algorithm to solve the associated optimization problem and determine the resource allocation / trunk size for each multicast group.

In terms of our QoE utility function, we need to determine the trunk size so we can determine each group's corresponding average bitrate and buffering ratio. QoE-aware resource allocation algorithms attempt to maximize their system utility in order to obtain the optimal user QoE per channel.

In the next section we discuss our proposed MAC-layer resource allocation algorithms.

III. Proposed Scheme

In order to evaluate video streaming QoE metrics, we must know whether a video frame is decodable or not. The typical unit used to evaluate a video streaming QoE metric, a video frame, exists at the application-layer. In an LTE system, frames, subframes, and slots are used to transmit data. Due to the limited capacity of the assigned RBs in a LTE slot, a video frame may require multiple slots in order to be completely transmitted. The capacity of a RB is dependent on its MCS and experienced channel conditions (channel conditions vary over time.) If the channel coherence duration is less than the duration of a video frame, the eNB should manage resources / assign RBs at the MAC-layer in order to make more efficient use of said RBs. Similarly, in order to perform QoE-based resource allocation, the QoE must also be evaluated at the MAC layer. Thus, in order to perform LTE slot resource allocation, we propose a MAC layer QoE utility function. This function is implemented as an Integer Linear Programming (ILP) resource allocation algorithm and is also implemented as a Gradient-based resource allocation algorithm.

In the first subsection we describe the formulation of the QoE function at the MAC layer. In the next subsection we discuss how the MAC layer QoE formulation can be structured as an ILP problem. Finally, in the last subsection, we discuss how the MAC layer QoE function can interpreted as a gradient and solved. Algorithms for both the ILP and Gradient approaches are also presented, in their respective subsections.

A. MAC-Layer QoE Utility Function

Based on the current video frame information and channel conditions, we *predict*[2] the trend of the QoE and assign RBs such that the predicted QoE is maximized.

In order to predict the QoE in each round of resource allocation we take the linear QoE utility, shown in equation (1), from an application layer QoE function and reformulate it as a MAC layer QoE function. Furthermore, we expand the definition of the linear function used in [11] to account for the multiple video layers associated with SVC. The linear QoE function is used as it requires less time to calculate than the exponential QoE function. For performance evaluation purposes, we use the more accurate exponential QoE utility, which is shown in equation (2).

We can reformulate the linear application layer QoE function, equation (1), into its MAC layer equivalent by examining how each variable contributes to the utility when the units under consideration are in terms of MAC allocation time periods versus video frames.

In regards to the buffering ratio *BuffRatio* (The buffering ratio is the percent of video frames which are dropped) and MAC-allocation period (T represents the MAC allocation period / scheduling interval. The allocation period can be adjusted by setting T equal to the number of slots. We can interpret a buffering event as being the result of an insufficient number of bits being allocated for the successfully transmission of the current video frame's base layer. A buffering event can be interpreted as an insufficient number of bits being allocated during the L allocation periods which comprise the transmission window, i.e., video frame deadline (An LTE frame is comprised of 10 subframes. Each subframe is comprised of 2 LTE slots. As each LTE frame is 10 milliseconds in duration, each LTE slot is 0.5 milliseconds in duration. Therefore, there are 2000 LTE slots per second. As such, for a given Frames Per Second (FPS), $L = \frac{2000}{FPS}$, i.e., for $FPS = 30$, a video frame is approximately $\frac{2000}{30} = 66$ LTE slots long in duration.) The number of remaining bits required to successfully transmit the base layer is $b^{[i]} = b^{[i]}_{F_{now,1}} - Bits^{[i]}$ (For video $[i]$, $b^{[i]}_{F_{now,\mathcal{L}}}$ represents the number of bits associated with the layer $\mathcal{L}$ in the current video frame F_{now}, and

[2] Predict here means we assume the channel will maintain roughly the same conditions for the duration of the TTI.

$Bits^{[i]}$ represents the number of bits already transmitted.) In every MAC-allocation period T the allocated trunk size $S^{[i]}$ is fixed. As such, we can characterize the probability of a buffering event occurring in any given allocation period as $P[Buff] = (1 - \frac{1}{u^{[i]}})^{+}$. The probability of a utility gain being obtained during the current allocation period, due to reducing the buffering ratio, is:

$$\frac{1}{u^{[i]}} = \frac{S^{[i]}}{b_{F^{[i]}_{now,\mathcal{L}}}} = \frac{S^{[i]}}{b_{F^{[i]}_{now,1}}} \tag{5}$$

Note: only the base layer has an effect on $u^{[i]}$ (The base layer is indicated by $\mathcal{L} = 1$.) As the utility has not yet been realized, the total number of bits $b^{[i]}_{F_{now,1}}$ is used, as opposed to the remaining number of bits $b^{[i]}$. Thus the utility gain is the probability that the current trunk size $S^{[i]}$, relative to $b^{[i]}_{F_{now,1}}$, is the last required trunk to complete transmission of the current layer. When $S^{[i]} \geq b^{[i]}_{F_{now,1}}$, $P[Buff] = 0$.

As such, the total number of potential buffering events is:

$$BuffEvents = N^{[i]}_{buff} + \left(1 - \frac{1}{u^{[i]}}\right) \tag{6}$$

where $N^{[i]}_{buff}$ represents the number of buffering events incurred by prior video frames, i.e., the number of incomplete transmissions.

The resulting buffering ratio is therefore:

$$BuffRatio = \frac{L}{t_{now} + T} \times BuffEvents \tag{7}$$

where t_{now} represents the duration of time since the epoch of the live video transmission.

In regards to the average bitrate, the average bitrate is:

$$AvgBitrate = \frac{Bits^{[i]} + S^{[i]}}{t_{now} + T} \tag{8}$$

where total number of bits already transmitted for video $[i]$ is $Bits^{[i]}$.

The MAC-layer formulation of the linear QoE function, from equation (1), is:

$$U^{[i]} = RE^{[i]} + DR^{[i]} \tag{9}$$

where $RE^{[i]}$ represents the effect the buffering ratio has on the utility and $DR^{[i]}$ represents the effect the data rate has on the utility, both of which are defined below.

$$\begin{aligned} RE^{[i]} &= \frac{w_{re} \times L}{t_{now} + T}\left(N^{[i]}_{buff} + \left(1 - \frac{1}{u^{[i]}}\right)\right) \\ &= \frac{w_{re} \times L}{t_{now} + T}\left(N^{[i]}_{buff} + 1 - \frac{T \sum_{j=1}^{N_{RB}} X^{[i]}_j M(m^{[i]}_j)}{b^{[i]}_{F_{now,1}}}\right) \end{aligned} \tag{10}$$

$$\begin{aligned} DR^{[i]} &= \frac{w_{kb}}{(t_{now} + T)}\left(kbits^{[i]} + \frac{S^{[i]}}{1000}\right) \\ &= \frac{w_b}{t_{now} + T}\left(Bits^{[i]} + T \sum_{j=1}^{N_{RB}} X^{[i]}_j M(m^{[i]}_j)\right) \end{aligned} \tag{11}$$

Note: For the data rate $DR^{[i]}$, the data rates $kbits^{[i]}$ and $Bits^{[i]}$ represent the aggregate data rates up to t_{now}. For the buffering ratio $RE^{[i]}$, only the base layer has an effect (the base layer is indicated by $\mathcal{L} = 1$.)

In the above formulation, and per table I, w_{re} represents the fractional utility loss associated with a buffering event. w_{kb} and w_b represent the fractional utility gain from a bitrate increase (w_{kb} is defined in terms of kilobits per second (kbps), while w_b is defined in bits per second (bps).)

In the next subsection we discuss the ILP formulation of the MAC-layer QoE function.

B. Integer Linear Programming Resource Allocation

In equation (12) we derive a system utility U over the set of video groups. Each group's contribution to the system utility is defined with respect to each group's resource block allocation $X^{[i]}_j$ and its corresponding weight function $g(n)$ (The weight function $g(n)$ is a group size weighted function which can be tuned by the network operator as per [16]; thus the concept of fairness can be accounted for based on the distribution of each group's size.)

$$U = \max_{X^{[i]}_j} \sum_i g(n) U^{[i]} \tag{12}$$

where $g(n)$ is per equation (13), and $U^{[i]}$ is per equation (9).

The weight function $g(n)$ is defined as:

$$g(n) = \begin{cases} 1 & \text{(Constant)} \\ log(n) & \text{(Logarithmic)} \\ n & \text{(Linear)} \end{cases} \tag{13}$$

The specific weighting function on the right hand side is indicated in the parenthesis. The specific weighting function is selected prior to simulation. Fairness, relative to group size, can be determined by selecting a different weight function [16]. Note: for the rest of the paper we set $g(n) = g(N^{[i]})$.

In order to formulate our problem as an Integer Linear Programming (ILP) problem, we apply the following 3 constraints:

$$\sum_{i=1}^{V} X^{[i]}_j = 1, \forall j \tag{14}$$

$$S^{[i]} \leq b^{[i]}_{F_{now,\mathcal{L}}} - Bits^{[i]} \tag{15}$$

$$X^{[i]}_j \in \{0, 1\}, \forall i, \forall j \tag{16}$$

The constraints (14) and (16) limit each channel to a single video at a time, where $X^{[i]}_j$ is an indicator variable. The constraint (15) places an upper-limit on the trunk size allocated to each video. As the over-allocated RB(s) would go unused, the upper-limit prevents the Service Provider from over-allocating RBs.

Using the aforementioned constraints and the system utility/objective, from equation (12), we use ILP plus branch-and-bound to obtain the allocation pattern $X^{[i]}_j$ in P-time [11]. As a result our solution is able to operate fast enough to be used for live video. We call our resource allocation algorithm the ILP Resource Allocation Algorithm, see Algorithm 1. We provide a detailed complexity analysis in Section IV.

Algorithm 1 ILP Resource Allocation Algorithm

```
Input: $T$: Allocation-Period, number of slots
 1: $t$: Time
 2: $m_j^{[i]}$: Estimated Channel Condition
 3: $g(N^{[i]})$: Weighted number of users in each video group
 4: $N_{RB}$: Number of Carriers / RBs
 5: $b_{F_{now,\mathcal{L}}}^{[i]}$: Packet-Size of current video frame, all layers
 6: $Bits^{[i]}$: Current amount of transmitted bits
 7: $V$: Number of Video Groups
 8: $\mathcal{L}_{max}^{[i]}$ : maximum number of layers per video group
 9: $\mathcal{L}^{[i]}$ : current layer number for video group
10: $FPS$: Frame Rate
Output: Channel Allocation Pattern, $X_j^{[i]}$
11: $L = 2000/FPS$, number of LTE slots per frame
12: $s_r = 100$, need to scale to percentage
13: $T_{sf} = 0.5$, scale factor, weights need to be slot relative
14: $A_{eq} =[\ ]$, $b_{eq} = [1]_{N_{RB}\times 1}$
15: for $i = 1$ to $V$ do
16:     $A_{eq} = [A_{eq}\ I_{N_{RB}\times N_{RB}}]$ //$I$ is an identity matrix
17: end for
18: for $i = 1$ to $V$ do //$b^{[i]}$ is an $V \times 1$ vector
19:     $\mathcal{L}_{base} = 1, \mathcal{L}_k = 0$
20:     while $\left(\sum_{k=1}^{\mathcal{L}_{base}+\mathcal{L}_k} (b_{F_{now,k}}^{[i]}) - Bits^{[i]}\right) \leq 0$ do
21:         $\mathcal{L}_k = \mathcal{L}_k + 1$
22:         if $(\mathcal{L}_k + \mathcal{L}_{base}) > \mathcal{L}_{max}^{[i]}$ then
23:             break
24:         end if
25:     end while
26:     $\mathcal{L}^{[i]} = \mathcal{L}_{base} + \mathcal{L}_k$
27:
        $b^{[i]} = \begin{cases} \sum_{k=1}^{\mathcal{L}^{[i]}} \left(b_{F_{now,k}}^{[i]}\right) - Bits^{[i]} & \text{when } \mathcal{L}^{[i]} \leq \mathcal{L}_{max}^{[i]} \\ 0 & \text{otherwise} \end{cases}$
28: end for
29: for $i = 1$ to $V$ do
30:     Calculate MCS to Bits Per Slot Mapping $M(m_j^{[i]})$
31:     $A_j^{[i]} = M(m_j^{[i]}) \times T$
32:
        $C^{[i]} = \begin{cases} T\left[\frac{-w_{re}\times s_r\times L}{(t+T)b_{F_{now,1}}^{[i]}} + \frac{w_b\times T_{sf}}{t+T}\right] & \text{when } \mathcal{L}^{[i]} = 1, \\ T\left[\frac{w_b\times T_{sf}}{t+T}\right] & \text{otherwise} \end{cases}$
33:     $f_j^{[i]} = g(N^{[i]}) \times C^{[i]} \times M(m_j^{[i]})$
34: end for
35: //Run ILP + Branch-and-Bound to solve $X_j^{[i]}$
    //ILP maximizes $f$ s.t. constraints (14), (16), and (15).
36: Return $X_j^{[i]}$
```

In the first stage of the algorithm we run multiple rounds over each RB in order to determine the utility associated with each video. In each round, after obtaining the channel conditions and the video packet information for a video, we tentatively allocate the RBs in order to determine each video's potential QoE function. After all of the rounds have finished, in the second stage of the algorithm, the QoE utility function from each video is used by an Integer Linear Programming solver to determine how the eNB should allocate RBs such that the aggregate set of the users' QoE is maximized. The proposed algorithm is summarized in Algorithm 1.

Note: When $\mathcal{L}_{max}^{[i]} = 1$, i.e., the maximum number of layers is 1 and the algorithm presented here is the same as in [11]. As we calculate the objective function on a per slot basis, the contribution of any given resource block to the objective function is dependent on the coefficients $C^{[i]}$ from equation (9), channel conditions $m_j^{[i]}$, and the weighted group size $g(N^{[i]})$. On line 33, the objective function $f_j^{[i]}$ is shown. The objective function's coefficient $C^{[i]}$ can be obtained by disregarding the terms used to collect buffering events $N_{buff}^{[i]} + 1$, the total number of bits already transmitted $Bits^{[i]}$, in addition to the bits per resource block $m_j^{[i]}$, and allocation term $X_j^{[i]}$. The $C^{[i]}$ term only needs to account for the incremental change in the objective function, while the terms $N^{[i]}$ and $M(m_j^{[i]})$ represent the group size and the number of bits for a resource block, respectively. The number of bits per resource block, given the allocation period T, is accounted for via the inequality constraint $A_j^{[i]}$. The remaining number of bits to be transmitted is accounted for via the inequality constraint $b^{[i]}$. The equality constraint A_{eq} permits any video group to be assigned any RB, while b_{eq} requires all RBs be assigned. For readability, we also add the term $s_r = 100$ as we would like to emphasize that the buffering ratio is a percentage.

In Algorithm 1, line 35, each video frame's layer's size is strictly enforced (a RB will not be assigned to a video if such an assignment causes the video layer's size to be exceeded) until a solution cannot be obtained, at which point the video layer's size limit is relaxed. After the video layer's size limit is relaxed, the ILP procedure will be re-run. This process continues until a solution is found for all videos, or all V videos are completely transmitted.

In the next subsection we define the Gradient-based resource allocation algorithm, which is based off the ILP algorithm.

C. Gradient-Based Resource Allocation

By observing that the value/capacity of the j^{th} RB is only dependent on the MCS used by the i^{th} group, in addition to the fact that each group's valuation of a RB is independent of each other group's valuation for said RB, the QoE resource allocation problem can be formulated as a gradient-based (GR) resource allocation problem. In the GR resource allocation problem, the direction of maximum increase in utility is also the optimal resource allocation, i.e., each RB is allocated to the group who values it the most. In this sense, the Greedy choice / locally optimal choice is always made.

Like ILP, in GR each video frame's layer's size limit is also strictly enforced. We use equation (12) and constraints (14)-(16) to formulate our problem. Unlike ILP though, we remove constraint (15) as a constraint. Instead we use equation (15) to strictly enforce the maximum value of the incremental increase in the utility function. This is achieved by adhering to the per video layer size limit. The insight is that there is no potential utility gain / value to be gained from any portion of a RB assignment which exceeds the video's layer's size limit. The maximum value / valuation of the incremental increase associated with a given RB assignment is:

$$f_{limit} = \begin{cases} f_j^{[i]} \frac{b^{[i]} - Bits^{[i]}}{M\left(m_j^{[i]}\right)} & 0 < (b^{[i]} - Bits^{[i]}) < M\left(m_j^{[i]}\right) \\ f_j^{[i]} & \left(b^{[i]} - Bits^{[i]}\right) \geq M\left(m_j^{[i]}\right) \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

As such, it is possible that some RBs will be assigned to a group which cannot fully utilize said RB, i.e., a RB assignment may occur despite the fact that the bitrate allocation for said layer would be exceeded. It should be noted that in ILP any bitrate allocation exceeding the required bitrate is delayed until all other video groups have their requirements met, i.e., the assignment can only occur when the bitrate constraints are relaxed (bitrate constraints are relaxed when all other groups have finished or are within 1 RB of finishing).

The Gradient-based algorithm can be constructed by using Algorithm 1 and replacing lines 35 to 36 with Algorithm 2.

In the next section we discuss our proposed and reference algorithms.

Algorithm 2 Gradient-Based Resource Allocation Algorithm

```
Input: $N_{RB}$: Number of Carriers / RBs
$V$: Number of Video Groups
$f_j^{[i]}$: Utility function for each RB $j$ and video group $[i]$
$b^{[i]}$: Packet-Size of current video frame for current layer
$Bits^{[i]}$: Current amount of Transmitted Bits
$m_j^{[i]}$: Estimated Channel Condition
Output: Channel Allocation Pattern, $X_j^{[i]}$
for $j = 1$ to $N_{RB}$ do
    $f_{max} = 0, RB_{assign} = 0$
    for $i = 1$ to $V$ do
        $m = M(m_j^{[i]})$
        $f_{limit} = \begin{cases} f_j^{[i]} \frac{b^{[i]} - Bits^{[i]}}{m} & 0 < (b^{[i]} - Bits^{[i]}) < m \\ f_j^{[i]} & (b^{[i]} - Bits^{[i]}) \geq m \\ 0 & \text{otherwise} \end{cases}$
        if $f_{limit} > f_{max}$ then
            $f_{max} = f_{limit}, RB_{assign} = i$
        end if
    end for
    if $RB_{assign} \neq 0$ then
        $i = RB_{assign}$
        $Bits^{[i]} = Bits^{[i]} + M(m_j^{[i]})$
        for $i = 1$ to $V$ do
            $X_j^{[i]} = \begin{cases} 1 & i = RB_{assign} \\ 0 & \text{otherwise} \end{cases}$
        end for
    end if
end for
Return $X_j^{[i]}$
```

IV. Description of Algorithms

In this section we compare our proposed resource allocation algorithms, Integer Linear Program (ILP) and Gradient (GR), against four well-known resource allocation algorithms. (The well-known resource allocation algorithms are: Baseline, Round Robin, Throughput-Oriented, and Water-Filling [28].) All algorithms transmit the video layers starting from the base layer up to the highest enhancement layer. All resource allocation algorithms are summarized below:

- **Baseline (BL):** This algorithm is a population-based proportional fairness scheduler resource allocation scheme. Each group i is allocated $\lfloor N_{RB} \times \frac{N^{[i]}}{\sum_{k=1}^{V}(N^{[k]})} \rfloor$ RBs. This allocation is fixed for the entire duration of the simulation. This algorithm treats all SVC layers within a group with equal importance and does not use a QoE-based utility function.
- **Round Robin (RR):** Every TTI, this algorithm allocates all RBs to a different group. If there are N_{RB} RBs and V groups, then on average every group will be allocated $\frac{N_{RB}}{V}$ RBs during the duration of the simulation. This algorithm treats all SVC layers within a group with equal importance and does not use a QoE-based utility function.
- **Throughput-Oriented (TO):** RBs are allocated with the objective of maximizing the aggregate system throughput. As such, larger groups with a given MCS are allocated RBs prior to smaller groups with the same MCS. This algorithm treats all SVC layers within a group with equal importance and does not use a QoE-based utility function.
- **Water-Filling (WF):** This algorithm refers to the RB allocation used in [28]. In order to make a fair comparison with our algorithms, we set each group's demand for their video stream to be equal to $\frac{b^{[i]}_{F_{now},\mathcal{L}}}{log_{10}(1+N^{[i]}) \times M(m^{[i]}_{j,k})}$. Each group's demand determines its relative priority (an interpretation of this scheme is the lower the demand is, the higher the priority is.) In other words, a video with small packet sizes, good channel conditions, or with a large group size will have a higher priority in terms of resource allocation. Every TTI, RBs are assigned to the video groups in the order of their respective priorities. Compared to [28], we modified the algorithm to account for the data size associated with each different video group, i.e., number of bits per video layer per video frame. As this algorithm differentiates between different video flows at the video layer level, it accounts for different SVC layers. This algorithm does not use a QoE-based utility function.
- **Integer Linear Programming (ILP):** Every TTI, this algorithm allocates RBs according to the each group's size, channel conditions, video packet sizes, and the QoE Utility function given in equation (12) per the constraints shown in equations (14), (16), and (15). See Section III for more details. The RBs are assigned such that the QoE utility is maximized. As this algorithm differentiates between different video flows at the video layer level, it

accounts for different SVC layers. This algorithm also makes use of a QoE-based utility function, thus it can optimize the user engagement metric.

- **Gradient-Based (GR):** This algorithm is similar to the ILP algorithm, but less complex. Every TTI, this algorithm allocates RBs according to each group's size, channel conditions, video packet sizes, and the QoE Utility function given in equation (12) per the constraints shown in equations (14) and (16), we remove the constraint shown in equation (15). Instead of using equation (15) as a constraint, we use it to limit the maximum value of the utility function, see Section III for more details. The RB allocation heuristic used assigns each RB to the group who offers the highest potential utility gain, per equation (17), i.e., this algorithm efficiently allocates each RB to the video / group who values it the most. As this algorithm differentiates between different video flows at the video layer level, it accounts for different SVC layers. This algorithm also makes use of a QoE-based utility function, thus it can optimize the user engagement metric.

In the next section we discuss the limitations of our proposed algorithms and how we fairly compare them.

A. Limitations of ILP

ILP results in an optimal resource allocation, yet to do this it requires a piecewise function to represent multiple layers via a utility function. Due to limitations of Matlab's Mixed Integer Linear Programming (MILP) solver, and the 0-1 integer programming problem, we can only optimize over a single utility function per TTI. As such, for a fair comparison against algorithms which are either QoE-aware and/or SVC-aware, we only consider a single SVC layer during each TTI.

Complexity Analysis is discussed in the next subsection.

B. Computational Complexity Analysis

In this section we discuss the computational complexity of each resource allocation algorithm.

The computational complexity of the BL algorithm is $O(V)$, where V represents the number of video groups. The distribution of RBs is dependent only on the population of all the groups, i.e., the channel conditions are not factored into this resource allocation algorithm. As the group sizes are considered constant-size, per scenario, this resource allocation scheme only needs to be run once during the simulation.

The computational complexity of the RR algorithm is also $O(V)$. In the RR algorithm the population of each group is not accounted for, only the number of groups is accounted for. Each resource allocation period, every TTI, a different group is allocated all of the RBs.

The computational complexity of the TO algorithm is $O(V * N_{RB})$, where N_{RB} represents the number of MBMS resource blocks. As the MCS of each RB is dependent on group channel conditions, each group must be examined to determine the allocation. This complexity is also the minimum computational complexity possible for any RB-level allocation algorithm. This algorithm is run once per TTI.

The computational complexity of both the GR and WF algorithms is $O(V * N_{RB} + V * \mathcal{L}_{max}) \approx O(V * N_{RB})$, where $\mathcal{L}_{max} = \underset{k \in V}{\arg\max}(\mathcal{L}^{[k]}_{max})$. To determine RB allocation, the MCS of each RB of each group must be examined; in addition, possibly every video layer must also be examined in order to determine which video layer is currently being transmitted. These algorithms are run once per TTI.

Integer Linear Programming, even 0-1 Integer Linear Programming, is a known NP-hard problem. From an exhaustive search standpoint, our ILP problem has a complexity which is in P-time, upper-bounded by $O((N_{RB})^V)$. Per theorem 1, ILP is an NP-hard problem.

Theorem 1: As the ILP problem can be reduced to a 0-1 Multidimensional Multiple-choice Knapsack Problem (MMKP), it is NP-hard.

Proof: From [29] we know that the any Integer Linear Programming problem can be reformulated as a Knapsack problem. For our particular problem, we can reformulate our ILP problem to a MMKP. The MMKP formulation of the problem is:

$$\max \sum_{j=1}^{N_{RB}} \sum_{i=1}^{V} f_j^{[i]} X_j^{[i]} \tag{18}$$

$$\text{subject to } \sum_{j=1}^{N_{RB}} M\left(m_j^{[i]}\right) X_j^{[i]} \le b^{[i]}_{F_{now,\mathcal{L}}}, i = \{1, \ldots, V\} \tag{19}$$

$$\sum_{i=1}^{V} X_j^{[i]} \le 1, j = \{1, \ldots, N_{RB}\} \tag{20}$$

$$X_j^{[i]} \in \{0, 1\} \tag{21}$$

As can be seen above, the single group per RB restriction can be interpreted as the Multiple-choice aspect of the problem, the set of N_{RB} RBs as the Multidimensional aspect of the problem, the value of a RB as per the objective function $f_j^{[i]}$, the weight of a RB as per the capacity of said RB $M(m_j^{[i]})$, and the group-specific capacity/layer size as $b^{[i]}_{F_{now,\mathcal{L}}}$, where $\mathcal{L}$ refers to the size of the current layer in the current video frame.

$\because$ MMKP is a known NP-hard problem.

$\therefore$ ILP is also a known NP-hard problem. ■

An issue with high computational complexity is the computation time takes longer, i.e., the TTI duration may need to be longer in order to permit the resource allocation algorithm time to finish. The problem with this is that increasing the TTI duration, decreases the accuracy of the CQI reports, i.e., there is an increased the risk of losing channel coherency. When a channel loses coherency, the resource allocation becomes inefficient, i.e., the performance becomes degraded [3].

In the next subsection, we discuss the differences between QoE approaches used by the ILP and GR algorithms.

C. Differences in QoE Optimization Approaches

Assuming the assignment of a RB does not exceed the remaining number of bits to be transmitted for group $[i]$ current video layer, i.e., $b^{[i]}$, both the GR and ILP algorithms are guaranteed to assign each RB to the video group which has the highest potential utility function gain from said RB. The difference between the two algorithms, as explained in Section III, is how the video frame's layer's size is enforced when making a RB allocation decision.

For ILP, as the resource allocation problem is solved via integer linear programming, each video frame's layer's size is strictly enforced. Initially, ILP will allocate a RB to the group which has the largest potential incremental contribution to the objective function, $f_j^{[i]}$, as long as doing so does not exceed the current video frame's layer's size limit. The problem with this approach is that layers which have less than 1 RB of bits remaining to transmit will not be assigned a RB until the size limit is increased (The size limit is only increased when no solution is found given the current size limits) or a RB of the appropriate size is allocated. As such, some layers may fail to be transmitted completely.

For GR, as each video frame's layer's size only determines the maximum value of the objective function $f_j^{[i]}$, it is possible that a partially used RB will be assigned to a group (a partially used RB indicates the group is less than 1 RB away from completing transmission of its current video layer.) As such, there will potentially be fewer partially transmitted layers. This approach permits the RBs to be assigned when realization of a video layer's contribution to the objective function / system utility is immediately possible.

As we only consider completely transmitted layers as having a contribution to the QoE utility function, we can see that in general ILP may have a lower utility than GR. As such, we can see that most of the time GR will result in an optimization which is as good as, if not better than, ILP's optimization. The exception is that GR may under-perform ILP when the potential utility gain from one group is substantially different from the potential utility gain from another group, i.e., the group's MCS, size, or the associated video layer's size substantially differs from that of another group.

In the next section we presented our simulation, results, and related analysis.

V. Simulation and Analysis

In this section we describe our simulator and simulation settings. Then we analyze our simulation results in terms of system performance, i.e., system utility.

A. Environment

The physical-level settings of our simulations are based on the LTE specifications [30] and [31]. The system-level settings are based on the LTE simulator [32].

As the LTE simulator doesn't contain an eMBMS simulation, we built our eMBMS simulator in Matlab. Using our simulator we compare the performance of our proposed resource allocation algorithms, ILP and GR, against 4 other commonly used resource allocation algorithms (The 4 other resource allocation algorithms are Baseline, Throughput-Oriented, Round Robin, and Water-Filling.) We summarize the simulation parameters and system model in table II.

TABLE II
Simulation Parameters and Models

Parameter	Setting
Symbols per slot	6 symbols (Extended Cyclic Prefix)
Number of Cells	Single Cell, neighbor cells act as interference
Cell layout	Hexagonal grid, 3 sectors/cell
ISD	1732m [TR36.814 Table A.2.1.1-1, case 3]
Central-frequency	2.0 GHz
System bandwidth	10 MHz
Locations of UE	Uniformly distributed in each sector
BS Power	46 dBm
Number of video streams	Simulation scenario dependent
LTE BLER upper-bound	10%
N0 (AWGN Noise)	Thermal noise density = -174dBm/Hz
Path loss and SF	UMa scenario in [TR36.814 Table B.1.2.1-1]
CQI definition	Per [TS36.213 Table 7.2.3-1]
Channel coherency duration	1 slot / 0.5ms
Allocation Period (TTI), T	1 slot / 0.5ms
Weight function $g(N^{[i]})$	$g(N^{[i]}) = N^{[i]}$ (Linear)

TABLE III
Test Video (Foreman, CIF @ 30 FPS) Settings

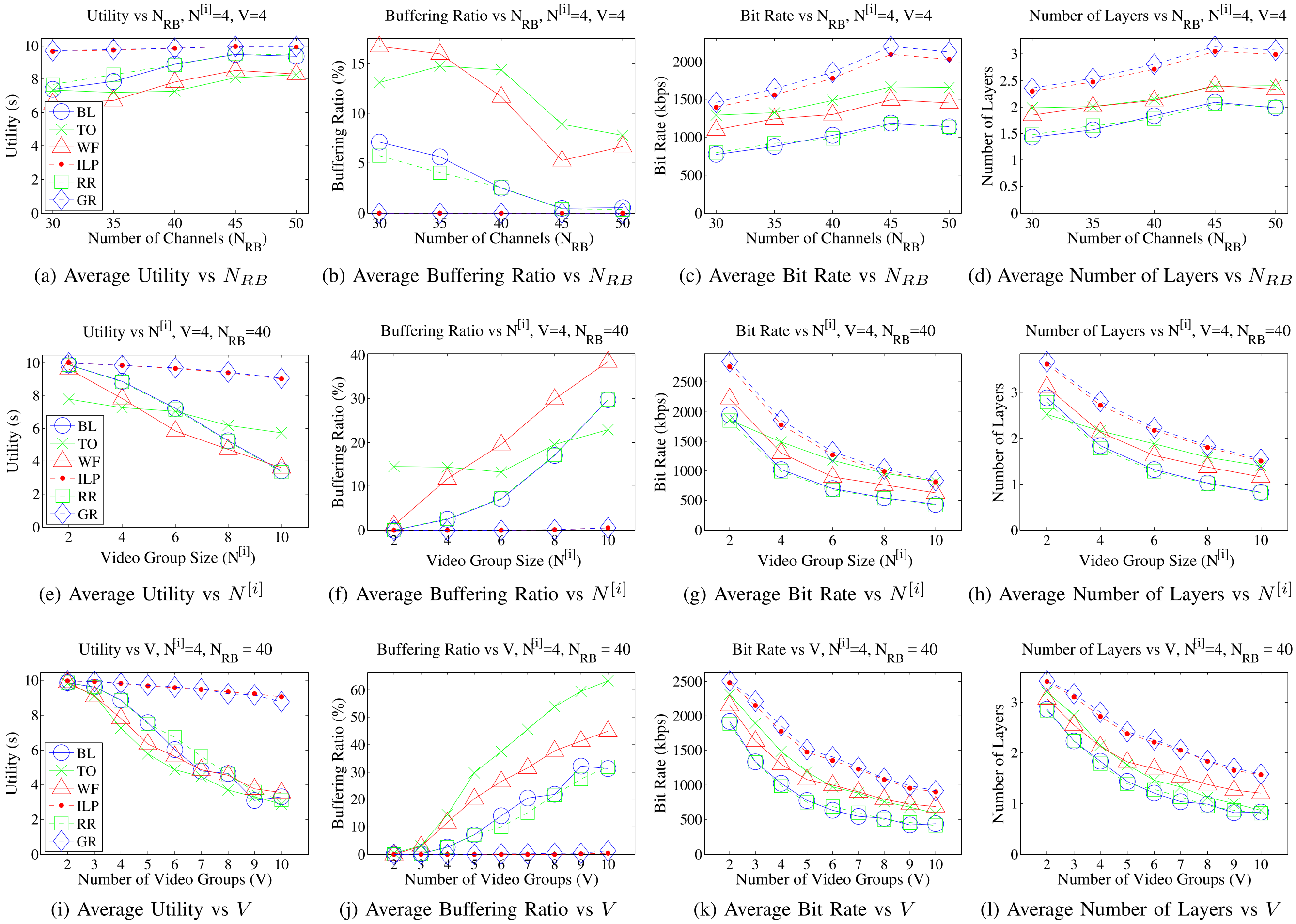
Layer Number (QP)	Average Bitrate
BL (40)	568.4 kbps
EL1 (36)	540.8 kbps
EL2 (32)	905.8 kbps
EL3 (28)	1356 kbps
Total	3371 kbps

We set our system bandwidth to 10MHz (per the LTE specification [31] there are 50 effective channels / RBs when the system bandwidth is 10MHz.) With the exception of the simulation scenarios where we vary the number of MBMS RBs, we assume 40 of the 50 channels are used for MBMS purposes, while the remaining channels are reserved for unicast or other applications. Other multicast configuration settings, such as the system population, are similar to those found in [16].

The video used for our simulation, Foreman [33], is in CIF format (352x288), has a duration of $D_i = 300$ frames, and a frame rate of 30 FPS. We encoded the video into 4 layers using JSVM [34]. The QPs and per layer and cumulative average bitrates are shown in table III.

Our system capacity is similar to Motorola's / AT&T's test scenario [35]. Due to the limitations of ILP, only a single layer may be transmitted per video per TTI. For a fair comparison against ILP we restrict the WF and GR algorithms to a single layer per TTI. Except for BL, each algorithm is run every TTI; BL is run only once, before the simulation begins. Every channel coherence period, i.e., slot, we recalculate the MCS of RBs and transmit the data based on the recalculated bandwidth. As such, given the resource allocation is subject to the present channel conditions, the simulation results show the maximum achievable utility function.

In the next subsection we present the simulation results and said analysis.



(a) Average Utility vs N_{RB} (b) Average Buffering Ratio vs N_{RB} (c) Average Bit Rate vs N_{RB} (d) Average Number of Layers vs N_{RB}

(e) Average Utility vs $N^{[i]}$ (f) Average Buffering Ratio vs $N^{[i]}$ (g) Average Bit Rate vs $N^{[i]}$ (h) Average Number of Layers vs $N^{[i]}$

(i) Average Utility vs V (j) Average Buffering Ratio vs V (k) Average Bit Rate vs V (l) Average Number of Layers vs V

Fig. 3. Performance Metrics for Different Scenarios.

B. Simulation Results and Analysis

The performance metrics in our simulation are presented below. Each metric is the averaged result over the set of simulations we performed. We simulated 17 different scenarios; each scenario was simulated 20 times.

- **Average Utility:** The expected utility / user engagement per user is evaluated per equation (2), the exponential form of the QoE utility function. The exponential form of the QoE utility function is used as it higher accuracy and helps ensure that the distortion due to the linear QoE does not adversely effect the results. Note: The utility may only increase upon the reception of a full video frame. The maximum utility is 10 (s), i.e., a video's maximum utility / maximum user engagement is defined as: $VideoLength = \frac{D_i}{FPS}$, where D_i is the duration of the video $[i]$ in terms of frames.
- **Average Buffering Ratio:** The average amount of time spent buffering divided over the entire video length.
- **Average Bitrate:** The average bitrate (kbps) per user.
- **Average Number of Layers:** The average number of layers received by the aggregate group. As the number of layers increases, the aggregate group's video quality also increases.

In figures 3a–3d we fix the group size and the number of groups, yet vary the number of MBMS RBs. As shown in figures 3a, 3c, and 3d, the aforementioned metrics improve when the number of available channels/RBs increases; yet as shown in figure 3b, the effect of increasing the number of channels has is less clear. Unlike the single layer SVC work [11], we see that WF and TO have the lowest utility and the highest buffering ratio. For WF, this is due to the fact that a given video group's enhancement layer is prioritized over another group's base layer; this can happen when the demand for the base layer is higher than that of an enhancement layer, i.e., due to channel conditions or relative layer sizes, the base layer's priority is lower than that of the enhancement layer's. For TO, groups which have a higher throughput will be assigned the relevant RBs over groups which suffer from lower throughput, i.e., groups with lower channel quality are more likely to be starved. As the base layer is relatively small relative to the entire set of video layers, see table III. RR and BL have a higher utility and lower buffering ratio. Both GR and ILP outperform all other algorithms, i.e., have the highest utilities / set of user engagement metrics, yet as we can see GR performs slightly better than ILP as it does not delay the assignment of a RB until all other groups are served. In figure 3d we can see the physical meaning of the QoE function, GR delivers the highest video quality over the aggregate set of groups. In general, as the average number of layers increases, so does the reconstructed video's quality/PSNR.

In figures 3e–3h, we fix the number of RBs and the number of groups, yet we vary each group's size (in each scenario each group has identical size.) As we can see in figure 3f, the algorithms which do not use a QoE-based utility function all have higher buffering ratios than those which use QoE-based utility functions. Similar to the single layer SVC metrics presented in [11], we see that the WF metrics become worse than the TO metrics as the group size increases, this is due to the fact that WF considers the demand of each video's current layer while TO does not, i.e., all else being equal, per table III, enhancement layer 1 gets prioritized over other video streams' base layer. Generally speaking as the population of a group increases, the group's MCS decreases, as determined per equation (4); thus the degradation in performance is not unexpected. Both channel quality aware algorithms TO and WF perform worse than other algorithms; this is due to the fact that groups with low MCSs will be starved of resources. Both GR and ILP outperform all other algorithms, i.e., have the highest utilities / set of user engagement metrics, yet as we can see GR performs slightly better than ILP. In figure 3h, we can see the physical meaning of the QoE function; GR delivers the highest video quality over the aggregate set of groups.

Finally, in figures 3i–3l, we show that the aforementioned metrics decrease as the number of groups increase, yet both ILP and GR outperform other algorithms. Unlike prior cases though, as the number of groups increase ILP outperforms GR. The reason for this is clear. ILP will delay assigning RBs to any group which cannot fully utilize them until all other groups which can fully utilize them are served. In this way, in a given TTI, groups with poorer channel quality conditions will be served prior to groups which require less than 1 RB to complete their transmission. As the number of groups increase, the delay of assigning a RB to a group which cannot fully utilize said RB likewise increases. When compared to ILP, GR obtains a higher bitrate and average number of layers; yet, as GR incurs a slight increase in its buffering ratio, it experiences a slightly lower utility, as shown in figure 3i, i.e., for groups with a size greater than 8.

In the next section we discuss how each algorithm compares.

VI. Discussion

While system performance, in terms of maximizing a utility function is important, there are other aspects to algorithmic design which should also be discussed. A system designer is also typically interested in knowing how an algorithm performs in terms of fairness, efficiency, and execution time. In this section we compare each algorithm in terms of its performance, fairness / efficiency, and execution time.

A. Fairness vs Efficiency

In this section we discuss how each algorithm behaves in terms of fairness vs efficiency. As these concepts are opposing, we emphasize which concept each algorithm is defined in terms of and how it attempts to achieve it.

In terms of fairness, we know the BL algorithm is a population-based proportional fairness algorithm, i.e., the amount of RBs assigned to a group is the ratio of its size relative to the total size of all the groups. The problem with this algorithm is not allocatively efficient,[3] i.e., does not account for the MCS of each RB, and is it QoE-aware; as such while the RBs are proportionally assigned, the bandwidth is not.

The RR algorithm allocates all RBs to a single group every TTI. This algorithm is fair in terms of resource utilization time, but is not allocatively efficient in terms of RB assignment as it does not evaluate the RB in terms of capacity. The problem with this algorithm is that groups which have low MCS will be starved due to a low bit rate, i.e., they may be unable to receive a video without experiencing an unduly high rate of buffering events.

The TO algorithm is a weighted population-based algorithm. This algorithm is not considered a fair algorithm, but is considered an allocatively efficient algorithm. Each RB is assigned to the group which has the largest aggregate bit rate increase associated with said RB. The problem with this algorithm is that groups which are either too small or do not have a sufficiently high MCS will experience a low bit rate, i.e., they may potentially be unable to receive the video without experiencing an unduly high rate of buffering events.

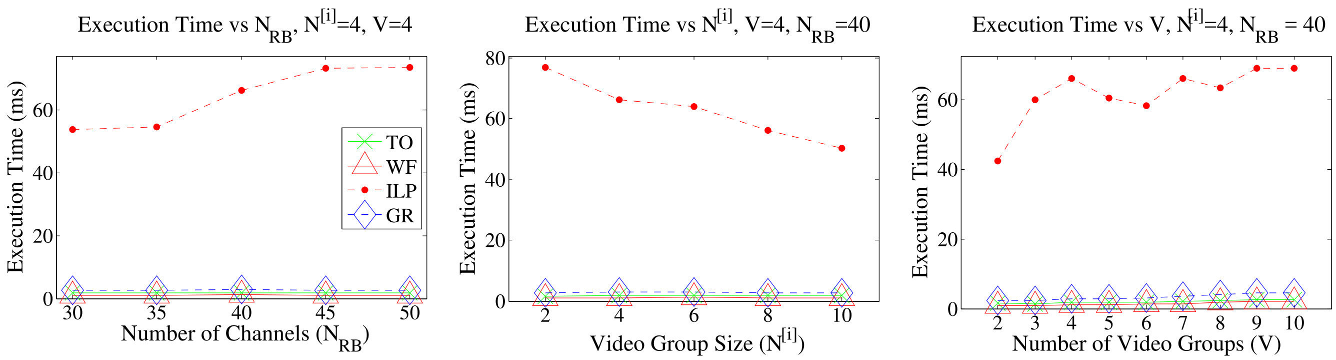
The WF algorithm is based on [28] and attempts to achieve Weighted Max-Min fairness.[4] This algorithm is not allocatively efficient as smaller demands are fulfilled at the expense of larger demands. In terms of video transmission, the problem with Max-Min fairness is that complex video frames/layers require a higher bitrate to be successfully transmitted. Thus all else being equal, the demand of said video frame/layer must necessarily be higher; consequently, a lower priority will be assigned to said video frame/layer. Similarly, videos associated with groups which have a lower MCS will also have a higher demand to transmit the same amount of data than similar videos associated with groups which have a higher MCS, i.e., the group with the lower MCS will be assigned a lower priority. Furthermore, this algorithm suffers from the scenario where an enhancement layer is smaller than a base layer, i.e., the demand for the enhancement layer is smaller than the demand for the base layer, as evidenced in the previous section and shown in table III. Thus when RBs are insufficient, more complex frames / groups with lower MCS may experience buffering events.

We know that the ILP and GR algorithms exhibit allocative efficiency, thus the RBs assigned to a group are determined by maximizing the aggregate QoE increase associated with said RBs, yet are not considered fair. All else being equal, buffering events will first occur on videos associated to groups which offer the smallest aggregate QoE increase per RB.

In the next subsection we discuss the execution of each algorithm.

[3] In the context of groups, an efficient allocation assigns an item/set of items to the group which values it the most [36].

[4] Weighted Max-Min fairness is achieved when it is impossible to increase the resource allocation to a flow with a larger demand without decreasing the allocation to a flow with a smaller demand.



(a) Average Execution Time vs N_{RB}

(b) Average Execution Time vs $N^{[i]}$

(c) Average Execution Time vs V

Fig. 4. Execution Time for Different Scenarios.

B. Execution Time Evaluation

As shown in figures 4a–4c, we can see that the two most significant factors impacting execution time are the number of RBs, N_{RB}, and the number of video groups, V (N_{RB} is limited by the bandwidth and MBMS configuration, while V is only limited by the MooD configuration.) As either the number of RBs or number of groups increases, we can see that ILP becomes less likely of being capable of real-time execution. The growth in ILP's execution time is substantially faster than that of other algorithms. While increasing the TTI duration would permit ILP more time to finish running prior to allocating RBs, the accuracy of the CQI reports decrease with time; as such, increasing the TTI duration also increases the risk of losing channel coherency. When a channel loses coherency, the resource allocation becomes inefficient, i.e., the performance becomes degraded [3].

We do not include the BL or RR resource allocation algorithms in the execution time analysis graphs as the BL algorithm is only run once, prior to any data transmission, and the RR algorithm only examines the number of groups / determines which group should get the entire set RBs for the current TTI, i.e., neither algorithm examines the individual state of the RBs. As such, neither mechanism is computationally intensive and can easily be shown to have a linear computational complexity.

In the next subsection we evaluate the performance of each algorithm.

C. Performance Evaluation

Based on our findings from the previous section and Section IV, we found that the commonly used resource allocation algorithms such as the population-based static BL, the group-based RR, the channel quality aware TO, and even the demand aware WF algorithms are all unable to efficiently allocate resources for a system transmitting SVC video. As the concept of SVC video and the importance of the base layer cannot be expressed to these algorithms, this resulting inefficiency was inevitable.

When we introduced QoE-aware resource allocation algorithms ILP [11] and GR, we found that not only did we outperform the other mechanisms in terms of user engagement / utility, but also in terms of video quality / number of layers delivered. This underscores the physical meaning of the QoE function as not only being about buffering ratio, but also being about video quality.

Furthermore, we found that in general we could improve the efficiency of ILP's resource allocation by relaxing the strict size limit associated with each video layer, as performed via the GR algorithm. Based on the findings presented in the previous section, we found that in most cases GR performs resource allocation as good as, if not better than, ILP. In the few cases where the GR resource allocation algorithm underperforms ILP, the loss is marginal.

Finally, based on a computational complexity analysis in Section IV, we found that the ILP has a substantially higher computational complexity than GR, i.e., NP-hard versus linear computational complexity. As such, it unlikely ILP will be used for real-time resource allocation. We also found that GR has similar computational complexity to WF and TO, i.e., all these algorithms have a linear computation complexity. While we know the BL and RR algorithms are even less computationally complex than TO and WF, neither of them consider the channel conditions; as such, neither of them is a viable candidate for maximizing user engagement or video quality.

In general, as GR performs equal to or better than ILP in terms of the aforementioned performance metrics and has a lower computational complexity than ILP, it is suggested that GR be used to perform real-time resource allocation.

VII. Conclusion

An on-demand resource allocation algorithm is a necessity due to the high volume of video traffic in LTE networks. Based on our simulation results we found that a QoE-based resource allocation algorithm achieves higher user satisfaction and video quality than traditional non-QoE aware resource allocation algorithms.

In this paper we proposed two QoE-Based resource allocation algorithms, GR and ILP, which efficiently allocate resources / RBs based on both the demand of the video and the channel conditions. Our algorithms are designed to maximize the QoE utility over the aggregate set of all users.

To test the algorithms we built an LTE eMBMS simulator whose environment is based on the LTE specifications [30] and [31]. The system settings are based on [32]. The system capacity is set per AT&T's / Motorola's settings [35].

We evaluated the performance of both of our resource allocation algorithms, GR and ILP, against 4 well-known resource allocation algorithms, TO, WF, BL, and RR. Our resource allocation algorithms always achieve the highest QoE utility, and the highest video quality, regardless of whether the resources / RBs are sufficient or not. When the number of groups is small, we found out that GR actually outperforms ILP. Only when the number of groups becomes large does ILP slightly outperform GR. In terms of computational complexity, we found that GR is significantly less computationally complex than ILP. Therefore, based on the performance and computational complexity analysis we suggest that the GR algorithm be used for real-time resource allocation. In our future work, we plan to investigate how pricing can be used to better allocate resources / RBs among video groups based on channel conditions and expected mobility. Groups which pay more could enjoy a higher video quality while not being restricted to slower speeds. Thus pricing could be used to differentiate service quality among video groups / create a tiered video service.

References

[1] C. V. N. Index, "The Zettabyte era—Trends and analysis," White Paper, Cisco, San Jose, CA, USA, 2013.

[2] 3GPP, "Multimedia broadcast/multicast service (MBMS) improvements; MBMS operation on demand," 3rd Gener. Partnership Project (3GPP), Sophia Antipolis, France, Tech. Rep. 26.489, Jun. 2015. [Online]. Available: http://www.3gpp.org/

[3] S. Sesia, I. Toufik, and M. Baker, *LTE—The UMTS Long Term Evolution From Theory to Practice*, 2nd ed. Chichester, U.K.: Wiley, 2011.

[4] R. O. Afolabi, A. Dadlani, and K. Kim, "Multicast scheduling and resource allocation algorithms for OFDMA-based systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 240–254, 1st Quart. 2013.

[5] H. Wang, H. P. Schwefel, and T. S. Toftegaard, "Adaptive modulation for a downlink multicast channel in OFDMA systems," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2007, pp. 650–655.

[6] H. Wang, H. P. Schwefel, and T. S. Toftegaard, "History-based adaptive modulation for a downlink multicast channel in OFDMA systems," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Las Vegas, NV, USA, Mar. 2008, pp. 1588–1592.

[7] R. Radhakrishnan, B. Tirouvengadam, and A. Nayak, "Channel quality-based AMC and smart scheduling scheme for SVC video transmission in LTE MBSFN networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Ottawa, ON, Canada, Jun. 2012, pp. 6514–6518.

[8] A. de la Fuente, A. G. Armada, and R. P. Leal, "Joint multicast/unicast scheduling with dynamic optimization for LTE multicast service," in *Proc. Eur. Wireless 20th Eur. Wireless Conf.*, Barcelona, Spain, May 2014, pp. 1–6.

[9] P. Li, H. Zhang, B. Zhao, and S. Rangarajan, "Scalable video multicast with adaptive modulation and coding in broadband wireless data systems," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 57–68, Feb. 2012.

[10] H. Schwartz, 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.

[11] H.-Y. Meng, C.-C. Chou, R. Kaliski, and H.-Y. Wei, "An on-demand QoE resource allocation algorithm for multi-flow LTE eMBMS," in *Proc. 24th Wireless Opt. Commun. Conf. (WOCC)*, Taipei, Taiwan, Oct. 2015, pp. 93–97.

[12] W.-H. Kuo, W. Liao, and T. Liu, "Adaptive resource allocation for layer-encoded IPTV multicasting in IEEE 802.16 WiMAX wireless networks," *IEEE Trans. Multimedia*, vol. 13, no. 1, pp. 116–124, Feb. 2011.

[13] Y. I. Choi, J. W. Kim, J. H. Kim, J. S. Jeong, and C. G. Kang, "Scalable transmission control: SVC-based dynamic resource allocation for enhanced multicast and broadcast service," *IEEE Commun. Lett.*, vol. 16, no. 9, pp. 1436–1438, Sep. 2012.

[14] L. Huang *et al.*, "Efficient group-based multimedia-on-demand service delivery in wireless networks," *IEEE Trans. Broadcast.*, vol. 52, no. 4, pp. 492–504, Dec. 2006.

[15] X. Wang, Z. Zhong, and Y. Zhao, "DeRe: A buffer saving and controllable video-on-demand broadcasting scheme for heterogeneous receivers," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 69–81, Mar. 2016.

[16] J. Chen *et al.*, "Fair and optimal resource allocation for LTE multicast (eMBMS): Group partitioning and dynamics," in *Proc. IEEE INFOCOM*, Apr. 2015, pp. 1266–1274.

[17] G. Araniti, M. Condoluci, L. Militano, and A. Iera, "Adaptive resource allocation to multicast services in LTE systems," *IEEE Trans. Broadcast.*, vol. 59, no. 4, pp. 658–664, Dec. 2013.

[18] A. E. Essaili, D. Schroeder, E. Steinbach, D. Staehle, and M. Shehada, "QoE-based traffic and resource management for adaptive HTTP video delivery in LTE," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 6, pp. 988–1001, Jun. 2015.

[19] T. Stockhammer, "Dynamic adaptive streaming over HTTP–: Standards and design principles," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst. (MMSys)*, Santa Clara, CA, USA, 2011, pp. 133–144. [Online]. Available: http://doi.acm.org/10.1145/1943552.1943572

[20] K. Piamrat, A. Ksentini, J.-M. Bonnin, and C. Viho, "Q-DRAM: QoE-based dynamic rate adaptation mechanism for multicast in wireless networks," in *Proc. IEEE GLOBECOM*, Honolulu, HI, USA, Nov. 2009, pp. 1–6.

[21] J. Koo and K. Chung, "MARC: Adaptive Rate Control scheme for improving the QoE of streaming services in mobile broadband networks," in *Proc. ISCIT*, Tokyo, Japan, Oct. 2010, pp. 105–110.

[22] H. Hu *et al.*, "QoE-based multi-stream scalable video adaptation over wireless networks with proxy," in *Proc. IEEE ICC*, Ottawa, ON, Canada, Jun. 2012, pp. 7088–7092.

[23] Y. Chen, K. Wu, and Q. Zhang, "From QoS to QoE: A tutorial on video quality assessment," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 1126–1165, 2nd Quart. 2015.

[24] F. Dobrian *et al.*, "Understanding the impact of video quality on user engagement," in *Proc. ACM SIGCOMM Conf. (SIGCOMM)*, Toronto, ON, Canada, 2011, pp. 362–373. [Online]. Available: http://doi.acm.org/10.1145/2018436.2018478

[25] X. Liu *et al.*, "A case for a coordinated Internet video control plane," in *Proc. ACM SIGCOMM Conf. Appl. Technol. Architect. Protocols Comput. Commun. (SIGCOMM)*, Helsinki, Finland, 2012, pp. 359–370. [Online]. Available: http://doi.acm.org/10.1145/2342356.2342431

[26] W.-H. Kuo, R. Kaliski, and H.-Y. Wei, "A QoE-based link adaptation scheme for H.264/SVC video multicast over IEEE 802.11," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 5, pp. 812–826, May 2015.

[27] S. Lu *et al.*, "Channel-aware frequency domain packet scheduling for MBMS in LTE," in *Proc. IEEE 69th Veh. Technol. Conf. VTC Spring*, Barcelona, Spain, Apr. 2009, pp. 1–5.

[28] C.-H. Ko, C.-C. Chou, H.-Y. Meng, and H.-Y. Wei, "Strategy-proof resource allocation mechanism for multi-flow wireless multicast," *IEEE Trans. Wireless Commun.*, vol. 14, no. 6, pp. 3143–3156, Jun. 2015.

[29] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Heidelberg, Germany: Springer, 2004.

[30] 3GPP, "Evolved universal terrestrial radio access (E-UTRA); physical layer procedures," 3rd Gener. Partnership Project (3GPP), Sophia Antipolis, France, Tech. Rep. 36.213, Sep. 2014. [Online]. Available: http://www.3gpp.org/

[31] 3GPP, "Further advancements for E-UTRA physical layer aspects; evolved universal terrestrial radio access," 3rd Gener. Partnership Project (3GPP), Sophia Antipolis, France, Tech. Rep. 36.814, Mar. 2010.

[32] J. C. Ikuno, M. Wrulich, and M. Rupp, "System level simulation of LTE network," in *Proc. IEEE 71st Veh. Technol. Conf.*, Taipei, Taiwan, May 2010, pp. 1–5.

[33] (2005). *SVC Test Sequences*. [Online]. Available: ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/

[34] K. Suehring. (Mar. 2012). *JSVM Reference Software*. [Online]. Available: http://www.hhi.fraunhofer.de/

[35] S. J. Crowley. (Apr. 2011). *The Challenge of HD Video Streaming on LTE*. [Online]. Available: http://stevencrowley.com/2011/04/22/streaming-hd-video-on-mobile-broadband/#more-1977

[36] N. Nisan, *Algorithmic Game Theory*. New York, NY, USA: Cambridge Univ. Press, 2007.

Rafael Kaliski received the B.S. degree in computer engineering and the M.S. degree in electrical engineering from California Polytechnic State University (Cal Poly), San Luis Obispo, in 2003 and 2005, respectively. He is currently pursuing the Ph.D. degree with the Graduate Institute of Electrical Engineering, National Taiwan University. After graduating from Cal Poly, he was with Cisco Systems Inc., for six years. He then studied Chinese with the International Chinese Language Program, National Taiwan University. His research interests include video coding, resource allocation, game theory, and networks.

Ching-Chun Chou received the B.S. degree in computer science and information engineering and the Ph.D. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2006 and 2016, respectively, where he is currently an Post-Doctoral Fellow. His research interests include wireless multicast networks, MAC protocol design, next-generation communication system simulation, and multimedia applications.

Hsiang-Yun Meng received the bachelor's degree in EECS from National Tsing Hua University, Hsinchu, Taiwan, in 2013, and the master's degree in electrical engineering from National Taiwan University, in 2016, His research interests include MAC protocol design in wireless networks. He is good at programming and applies this knowledge in wireless network. Starting from senior high school, he attended several computer programming contests. Since 2016, he has been with the System-Tool-Team of Wireless-Communication-Technology, MediaTek. He was a recipient of an Excellence Award from the National Collegiate Programming Contest of Taiwan, in 2010, and the Third Place Prize in the 2008 Taiwan National Senior High School Programming Contest. He has an honorable mention in the 2010 Asia Kaohsiung Regional Contest.

Hung-Yu Wei received the B.S. degree in electrical engineering from National Taiwan University (NTU), in 1999, the M.S. and Ph.D. degrees in electrical engineering from Columbia University, in 2001 and 2005, respectively. He was a summer intern with Telcordia Applied Research, in 2000 and 2001. He was with NEC Labs America, from 2003 to 2005. He joined the Department of Electrical Engineering, NTU, in 2005, where he is currently a Professor with the Department of Electrical Engineering and Graduate Institute of Communication Engineering. He was a Consulting Member of the Acts and Regulation Committee of the National Communications Commission from 2008 to 2009. He actively participates in wireless communications standardization activities, and was a Voting Member in the IEEE 802.16 working group. His research interests include broadband wireless communications, vehicular networking, cross-layer design for wireless multimedia communications, Internet of Things, and game theoretic models for networking. He was a recipient of the Recruiting Outstanding Young Scholar Award from the Foundation for the Advancement of Outstanding Scholarship in 2006, the K. T. Li Young Researcher Award from ACM Taipei Chapter and IICM in 2012, the CIEE Excellent Young Engineer Award in 2014, the NTU Excellent Teaching Award in 2008, the Research Project for Excellent Young Scholars from Taiwan's Ministry of Science and Technology in 2014, the Wu Ta You Memorial Award from Ministry of Science and Technology in 2015. He is currently the Chair of the IEEE Vehicular Technology Society Taipei Section. He also serves as an Associate Editor for the IEEE INTERNET OF THINGS JOURNAL.