This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3021229, IEEE Transactions on Multimedia

Optimizing Video QoE For Mobile eMBMS Users in Cellular Networks

Ahmed Khalid, Ahmed H. Zahran and Cormac J. Sreenan
Email: a.khalid@cs.ucc.ie, a.zahran@cs.ucc.ie and cjs@cs.ucc.ie
School of Computer Science, University College Cork, Ireland

***Abstract*—Evolved Multimedia Broadcast Multicast Service (eMBMS) is used in cellular networks to improve the utilization of scarce wireless resources in high user density service areas. However, eMBMS configuration involves interwoven decisions including which base stations (eNB) to synchronize to form Single Frequency Networks (SFN), which video qualities to be serviced, and how to distribute resources among different videos. These decisions should accommodate disparate channel conditions for eMBMS users and the impact of eNB's unicast-load in the service area. In this paper, we formulate eMBMS configuration as an optimization problem that maximizes the video QoE for users. Additionally, we present NIMBLE as an eMBMS configuration heuristic, guided by our optimization framework, to solve the problem in realtime. Furthermore, NIMBLE's design integrates elements to accommodate the dynamic nature of cellular networks resulting from changes in both user and network state over time. We developed a simulation testbed and performed extensive experiments to show that, in comparison to state-of-the-art schemes, NIMBLE can increase the average user throughput by 150% and reduce the bitrate switches by 75%.**

I. Introduction

The growing demand for bandwidth-intensive live streaming services [1] has made the spectrum allocation challenging for cellular network providers. With up to 6 billion mobile broadband subscriptions in 2019, user preferences have shifted towards watching videos over hand-held mobile devices [2] and the user expectation for video quality has increased making higher definition (HD) videos more prevalent [3]. The problem of resource allocation becomes further challenging when users in densely populated and crowded areas subscribe to live video streams and request the same content at the same time, increasing the peak bandwidth requirements. Using unicast transmission mode in such an environment leads to inefficient resource utilization and poor user experience.

Evolved Multimedia Broadcast Multicast Service (eMBMS) is a 3GPP standard [4] employed in 4G and 5G networks, that provides an efficient method for delivering live content to a large number of cellular network users. To improve the resource utilization, eMBMS allows sharing resources among groups of users watching the same content and transmitting the content once to every group. Furthermore, eMBMS allows base stations (eNBs) in a spatially local area to transmit a video synchronously at a common frequency and time (Figure 1). The synchronized eNBs create a cluster of base stations, commonly known as a Single Frequency Network (SFN). The synchronized content delivery in an SFN improves the channel condition of eMBMS users.

The ultimate goal of cellular operators is to maximize the quality of experience (QoE) for users by optimizing the utilization of the scarce wireless resources. This optimization involves key decisions for configuring the physical network for eMBMS. Additionally, it incorporates various factors including network-related factors such as channel conditions of video users and resources available across eNBs, and traffic-related factors such as available video bitrates and number of users per video. A key operator decision is to determine which eNBs should be synchronized to form one or more SFN clusters for each video.

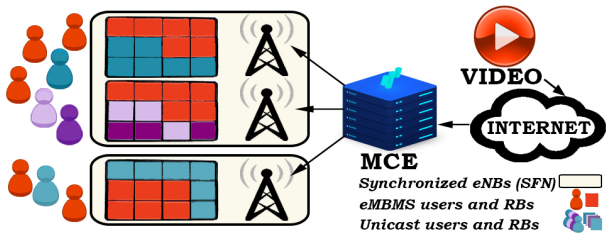


Fig. 1. eMBMS Architecture. *eNBs in a cluster transmit the content on same RBs and schedule their unicast users with the remaining RBs.*

Within an SFN cluster, users interested in the same video may have heterogeneous channel conditions. Resource management solutions should ensure that the transmitted video is decodable by all users. Placing all users in one multicast group is unfair to the users with good channel condition. Alternatively, creating multiple *user-groups*, where each *user-group* receives a different video quality can improve the overall QoE. However, creating too many *user-groups* fails to take advantage of multicast and reduces the cumulative throughput of the system [5]. Therefore, the number of *user-groups* represents another key decision when optimizing eMBMS resource management.

Finally, identifying the number of resources to allocate per *user-group* is another key decision to be made by an operator and it depends on the total available resources for eMBMS per cluster and the channel condition of every user. Note that eMBMS users have to share the OFDMA resources with non-eMBMS unicast users whose load may vary from one eNB to another and the impact of decisions made for eMBMS on unicast users must also be considered.

Several eMBMS resource management solutions have been proposed to solve the network configuration problem [5], [6], [7]. However, the existing algorithms are based on fairly-static network scenarios and do not consider the dynamics of the network state or mobile users. Additionally, evolving eMBMS standardization, such as MBMS Operation-On-Demand [8] (MOOD), and extension of eMBMS into 5G networks [9],

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3021229, IEEE Transactions on Multimedia

imply the need for solutions capable of accommodating network dynamics in real-time.

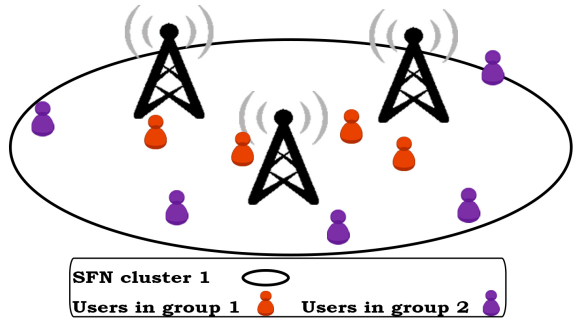
In this paper, we present a resource management framework that configures the physical network with the objective to maximize the end-user experience. Our proposed solution is extremely scalable and dynamic, making it feasible for deployment in real-world cellular networks. Our contributions in this paper are as follows:

- We formulate an optimization problem that maximizes the aggregate user experience by jointly determining the formation of SFN clusters, the user grouping, video resource allocation, and the bitrate selection. Furthermore, our novel model considers the three fundamental factors of QoE when allocating resources: the video bitrates received by users, video frames dropped or skipped by users, and the switches in video bitrates encountered by users.
- We propose a heuristics-based algorithm, NIMBLE, that solves the optimization model in real-time regardless of the number of users and their mobility profiles. NIMBLE also introduces *control* parameters for the network reconfiguration to reduce frequent video quality switches for users and overhead cost of network management. These parameters help further improve the QoE of end-users.
- We evaluate the solution using a simulation-based testbed for eMBMS and implement real-world scenarios with varying mobility patterns for users. We generate traces of real videos with different bitrates to analyze the behavior of NIMBLE and compare it with state-of-the art approaches [5], [6], [7]. We conduct extensive evaluation to show that NIMBLE boosts the average received video-bitrate by 150% and reduces the bitrate switches by 75% in comparison to the state-of-the-art solutions. These improvements lead to an increase in the QoE for more than 85% of the users.

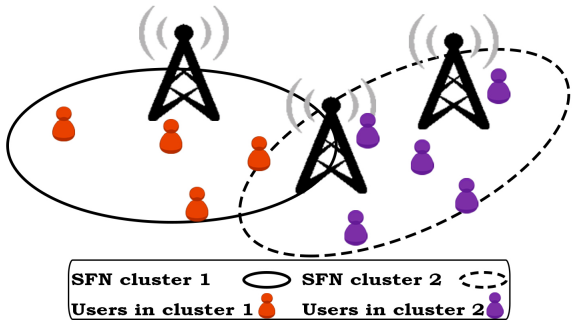
The rest of the paper is organized as follows. In Section II, we review background and literature on eMBMS. Section III presents the system model and formulates the global optimization problem while Section IV is dedicated to the proposed dynamic network reconfiguration algorithm, NIMBLE. Our performance evaluation setup and results are presented in Section V. Finally, Section VI presents our conclusions.

II. Literature Review

For live video streaming, multicast can be used instead of unicast to avoid unnecessary packet duplication. Content Delivery Networks (CDN) can provide a relief to the Internet backbone by using multicast to deliver content to the edge. For inter-domain content delivery, a large amount of literature focuses on using multicast at the network [10] or application [11] layers. This approach reduces resource consumption in the core and wired access network. However, the last hop in cellular networks is wireless, where the higher layer multicast gets converted back to unicast, resulting in redundant transmissions and wastage of physical resources.



a *Synchronize eNBs to form one cluster. Partition users into groups based on their channel condition.*



b *Create multiple SFN clusters (two in this case) based on user distribution and network state.*

Fig. 2. Examples of possible eMBMS configurations.

Studies that focus on unicast optimization, e.g. [12], cannot directly be extended and applied to multicast, due to the inherent differences in the two transmission mechanisms. Similarly, research conducted to improve energy [13] or utility [14] for TCP-based transport in wireless networks cannot be applied to eMBMS, as the preferred transport mode in eMBMS is UDP due to multicast transmission and live streaming. At the physical layer, eMBMS takes advantage of the inherent broadcast nature of wireless channel to allow sharing resources among users watching the same video stream and improves resource utilization.

In this section, we take a look at how eMBMS enables multicast at the physical layer. We then review the existing works on eMBMS. Finally we look at the QoE and fairness approaches to measure the performance of a delivery system.

A. Background on eMBMS

In cellular LTE networks (Figure 1), the unit of resource allocation is a resource block (RB), which represents the time and frequency at which a signal is transmitted. An eNB schedules its non-eMBMS unicast users by allocating them RBs, often through some variant of a proportionally fair scheduler [15]. While doing so, the eNB chooses a modulation and coding scheme (MCS) for each user based on their reported channel condition. Higher MCS yields higher spectral efficiency however, if a user is assigned an MCS higher than what its channel condition can support, it would be unable to decode the signal properly and experience packet losses.

For eMBMS [4], users can be combined into groups receiving various streams. These streams could be different videos or

the same video at different bitrates (Figure 2a). All *user-group* members receive the content over the same set of RBs. To ensure that all members of a *user-group* can properly decode the signal, the MCS of the *user-group* is restricted to the user with the worst channel condition. Therefore, to increase the system spectral efficiency, it is essential to create *user-groups* commensurate to the channel conditions of users.

eMBMS also facilitates creating SFNs in densely populated areas when multiple neighboring eNBs have to serve the same video. eNBs that are part of an SFN, transmit a video in a synchronous manner over the same set of RBs. Interested users combine the received signals, boosting their Signal to Interference-Noise Ratio (SINR) and hence, their spectral efficiency. Since different eNBs in an eMBMS service area may have different number of non-eMBMS users, placing an eNB with a higher unicast load in an SFN may limit the number of RBs available to eMBMS sessions. Hence, creating multiple SFN clusters in an eMBMS service area can be leveraged to improve eMBMS user achievable throughput (Figure 2b).

eMBMS management is handled by a centralized entity called the Multicast Coordination Entity (MCE). The MCE schedules resources for eMBMS users and instructs the involved eNBs to ensure synchronized transmission. The scheduling process involves defining *user-groups* and SFN clusters and assigning a certain number of RBs to each *user-group*.

B. Related Work on eMBMS

Several researchers have proposed resource management solutions for eMBMS. Hoe et al. [16] utilize users with good channel conditions as relays for users with bad channel conditions. Such methods can be hard to implement in real systems due to the greedy nature of users and low-latency requirements of live streaming. Muvi [17] uses scalable video coding in an attempt to maximize the utility for multicast users but does not consider the unicast network load. Won et al. [18] solves user-grouping problem for a single-cell network and assigns an MCS value to each *user-group* with the goal to maximize proportional fairness, but sometimes places users in groups where the MCS value is higher than what they can decode.

In [5], [19], [20], optimization models and algorithms are proposed that group users based on their channel conditions (Figure 2a) while considering the impact on unicast users. These models do not address SFN clustering problem and assume that all the eNBs in the service area always stay synchronized. Furthermore, [5] only solves the problem for a single eMBMS video. BoLTE [6] addresses the SFN clustering problem for multiple broadcast sessions (Figure 2b) but places all users in a single group, which is unfair to users with good conditions and limits the achievable utility.

Unlike some unicast-based resource allocation algorithms (e.g., [21]), another issue with aforementioned eMBMS solutions is that they try to maximize the network-level throughput rather than the application-level video bitrates. Consequently, they may allocate resources to all *user-groups*, but they cannot guarantee that the throughput is enough to receive an available bitrate of the transmitted video.

In our previous work RTOP [7], we proposed an algorithm that aims at maximizing a utility based on application-level user bitrates. RTOP also solves a joint optimization problem by addressing both user-grouping and SFN clustering and considers the possible variability in RBs across eNBs. However, RTOP does not consider switches in bitrates that users may experience due to the temporal dynamics of cellular networks, especially in mobile scenarios.

C. QoE and Fairness

The quality of experience measures the delight or annoyance of a user when watching a video. QoE is generally a subjective metric and measured with Mean of opinion score (MoS) [22]. However, several efforts have been made to define objective metrics which are helpful when designing algorithms and measuring or comparing performance of various algorithms. MIQX [23] uses active learning to model the interactions between quality of service factors and QoE.

In [24], authors conduct subjective tests to derive impairment functions for different temporal and visual QoE components and formulate an overall QoE model. Duanmu et al. [25] and Song et al. [26] utilize regression-based techniques and present QoE equations with high correlation to subjective user-QoE. These studies indicate that stalls could be the highest source of annoyance while the average video bitrate is highly correlated to the MoS.

To measure fairness among users with homogeneous underlying network capacities, Jain's index [27] uses variation coefficient of user throughput whereas f-index [28] uses standard deviation in QoE values along with lower and upper bounds. In a wireless environment however, where users may have disparate channel condition and maximum achievable throughput, the fairly shared spectrum efficiency (FSSE) is a common approach to jointly measure fairness and system spectrum efficiency. A widely used metric in wireless networks is Proportional Fairness [15] which aims to maximize the sum-log of user bitrates, often with variations to avoid scheduling starvation (i.e. some users allocated no spectrum).

III. Optimizing eMBMS User Experience

In this section, we formulate an optimization model that maximizes the QoE of system users while fairly allocating resources to users of multiple videos in an eMBMS service area. Table I summarizes the notations used in this section.

A. System Model

We consider an eMBMS service area including a set of identically configured eNBs, denoted by B. A set of videos, V, are served through eMBMS to users in set M. Each video is encoded to a set of bitrates, denoted by R_v. For each video $v \in V$, eNBs can be synchronized to form SFN clusters (denoted by set C). We denote a combination of non-overlapping SFN clusters, that cover the whole service area, as

an *SFN layout*. Depending on how clusters are created, various possible layout configurations are possible. For example, the layout in Figure 2a, is composed of one SFN cluster and the layout in Figure 2b is composed of two SFN clusters. We represent all possible SFN layouts for the eMBMS service area by set L. Note that each video may be served over a different layout configuration.

For a given layout l, a binary variable β_{blc} identifies whether an eNB b is placed in cluster c. Each eNB b has N RBs available and reserves Y_b RBs for its unicast users and dedicates the remaining RBs to eMBMS users. Operators can determine Y_b based on their priority of eMBMS over unicast [6] or through a multicast weight function that modulates the resource allocation between unicast and eMBMS users [5] e.g. by considering the number of users of each type. We assume that each eNB b provides Y_b as an input to the MCE. The number of RBs an eNB b can allocate to eMBMS is denoted by ρ_b and is calculated as the minimum of a fraction α [29] of total RBs (αN) and the RB balance after deducting the unicast share ($N - Y_b$); i.e.

$$\rho_b = \min(\alpha N, N - Y_b) \tag{1}$$

We assume that the reserved resources at every eNB, i.e., ρ_b, are sufficient to serve its eMBMS users.

When users report their reference signal received power (RSRP) based on cell-specific reference signals, eNBs determine the achievable spectral efficiency for their associated unicast users and allocates them resources accordingly. For every eMBMS user $m \in M$, the report is delivered to MCE that estimates the user's Signal-to-Interference ratio (SINR) from each possible SFN cluster and calculates the achievable spectral efficiencies E_{mc} $\forall c \in C$.

Based on the channel condition of users in an SFN cluster, the MCE may choose to transmit one or more distinct bitrates per video, each at a different MCS. We consider a standard eMBMS client [8] that may select a bitrate, and consequently an MCS, which is best suitable for its channel condition. The spectral efficiency of the chosen MCS, determines the number of RBs needed to achieve that bitrate.

We define a complete network configuration as an SFN layout l chosen for each video, bitrates transmitted by SFN clusters in l and the physical-layer parameters i.e., the MCS and number of RBs on which a bitrate is transmitted.

B. Problem Formulation

Our objective is to optimize the QoE of users by considering the key streaming components including bitrate and switching utilities as detailed below. Note that video stalls are avoided by ensuring that enough resources are allocated to users to seamlessly decode the streamed video quality.

Bitrate Utility of eMBMS users: For an SFN layout l, we define this component as the cumulative bitrate utility of all the eMBMS users M_v of video v in each cluster $c \in l$, and calculate it as:

$$Q(M_v) = \sum_{c \in l} \sum_{r \in R_v} \log(r) \sum_{m \in M_v} U_{mvlcr}, \tag{2}$$

TABLE I
NOTATIONS FOR OPTIMIZATION MODEL

Symbol	Description
INPUTS	
B	Set of one or more eNBs in the eMBMS service area
C	Set of possible clusters of eNBs (non-empty subsets of B)
L	Set of all possible SFN layouts i.e. ways to configure eNBs B into non-overlapping clusters
β_{blc}	Binary variable to inform if eNB b is in cluster c for layout l
N	Total number of resource blocks available at any eNB
α	Maximum fraction of resources allowed for eMBMS
Y_b	Number of RBs requested by eNB b for its unicast user u
V	Set of videos served by eMBMS in the service area
R_v	Set of bitrates available for video v
M, M_v	Set of multicast users (M) subscribed to video v (M_v)
E_{mc}	Spectral efficiency achievable by user m in cluster c
γ_v	Weight of bitrate-switching penalty for video v
r'_m	Current streaming bitrate of user m
VARIABLES	
L_{vl}	Binary variable to determine whether SFN layout l has been chosen for video v
U_{mvlcr}	Binary variable to determine whether a user m watching video v is placed in cluster $c \in l$ and assigned bitrate r
X_{vlcr}	Number of RBs allocated by eNBs in cluster $c \in l$ to video v for bitrate r

where U_{mvlcr} is a binary variable that decides whether a user m interested in video v located in cluster c of an SFN layout l should receive the bitrate r. The log function here serves two purposes. First, it reflects the fact that the marginal utility of video bitrate decreases as the rate increases [30]. Second, it is compatible with typical resource allocation policies in cellular networks, e.g., proportional-fairness (PF).

Switching Utility for eMBMS users: Switches in video bitrates is another factor that impacts the video-watching experience for users. With a layout l for a video v, the total switching-penalty of all the users M_v can be calculated as:

$$S(M_v) = \sum_{c \in l} \sum_{r \in R_v} \sum_{m \in M_v} f(m, r) \cdot U_{mvlcr}, \tag{3}$$

where $f(m, r)$ is a function that measures the impact of switching a user m from its current bitrate to bitrate r.

The impact of quality switching is also captured using a log-based utility and can be expressed as:

$$f(m, r) = \log(r'_m) - \log(r), \tag{4}$$

where r'_m is the current streaming bitrate of user m. The logarithmic function reflects the reduced impact of visual experience when switching between higher bitrates. In Equation 4, switching to a higher bitrate ($r > r'_m$) yields a negative value (i.e., reduced penalty or improved QoE) and switching to a lower bitrate ($r < r'_m$) yields a positive value (i.e., increased penalty or reduced QoE).

The importance of the switching component varies across QoE metrics [24]. While some QoE metrics consider the total number of switches in a session, others look at the switching magnitude [25] to capture abrupt quality variations. Some QoE metrics penalize switching both to a lower or a higher bitrate. For such metrics, the absolute value of the switching metric is typically used [25]. An operator can choose a metric that best

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3021229, IEEE Transactions on Multimedia

suits the demand of their users and $f(m, r)$ can be defined accordingly.

Another practical aspect is that identifying the switching utility implies the accurate knowledge of user's current bitrate r'_m. However, eMBMS does not have a feedback loop between a user's video-client and MCE, limiting the MCE's knowledge of current user-state. Establishing such a loop may incur high overhead cost and is usually avoided for broadcast/multicast based systems, such as eMBMS. Therefore, a mechanism is needed to estimate the switching penalty without a user's application-level feedback. Such an approach may not yield optimal results but is essential to enable real-world deployment of the optimization model.

To overcome this issue, we assume that the user would be streaming the highest bitrate being served in its SFN cluster and decodable with its channel condition (i.e. MCS). Note that the MCE is aware of both the current user channel-condition, as reported by the standard LTE module, and the streamed video bitrates in the user's cluster with the current network configuration. Hence, the switching utility for each user can be calculated for any next rate accordingly (Section IV-C).

Frames lost by eMBMS users: Generally, wireless users experience losses if their SINR is not high enough to properly decode the MCS assigned to them. Depending on how live video is streamed, these losses can result in video stalls, freezes or dropped and skipped frames. Such temporal quality degradation is considered the most annoying factor for users [31] and negatively impact the user QoE. To eliminate frame losses, we consider a constraint to our optimization model to ensure that the bitrate assigned to a user can be successfully received given its channel quality. Furthermore, the constraint ensures a stall-free transmission, by allocating sufficient resources to each transmitted bitrate.

Optimization model: Our QoE-oriented eMBMS resource optimization problem is formulated as follows:

Problem 1: *Maximizing QoE of eMBMS users*

$$\sum_{v\in V}\sum_{l\in L} L_{vl}\cdot\Big(Q(M_v)-\gamma_v\cdot S(M_v)\Big) \tag{5a}$$

subject to

$$\sum_{l\in L} L_{vl}=1,\ \forall v\in V \tag{5b}$$

$$\sum_{c\in l}\sum_{r\in R_v} U_{mvlcr}\le 1, \forall v\in V, l\in L \tag{5c}$$

$$\sum_{l\in L}\sum_{c\in l}\sum_{r\in R_v} U_{mvlcr}=M_v,\ \forall v\in V \tag{5d}$$

$$X_{vlcr}\cdot E_{mc}\ge r\cdot U_{mvlcr}, \forall m\in M, v\in V, r\in R_v, c\in l\in L \tag{5e}$$

$$\sum_{v\in V}\sum_{l\in L}\sum_{c\in l}\sum_{r\in R_v}\beta_{blc}\cdot X_{vlcr}\le\rho_b, \forall b\in B \tag{5f}$$

where L_{vl} is a binary variable to determine whether SFN layout l has been chosen for video v and X_{vlcr} is the number of RBs allocated to the bitrate r of video v in cluster $c\in l$. γ_v is the weight assigned to the switching penalty for video v and can be adjusted based on operator-defined utility or user mobility patterns.

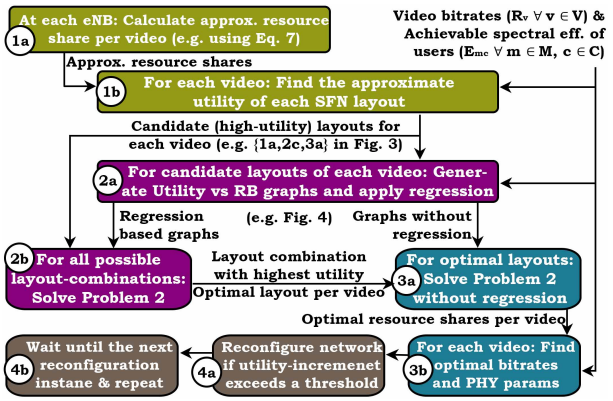


Fig. 3. Flow chart of the NIMBLE algorithm. *The utility is comprised of user-bitrates and switches.*

Constraint 5b ensures that one SFN layout is chosen for each video. Constraint 5c guarantees that a user is neither assigned more than one bitrate nor placed in more than one clusters. Constraint 5d enforces that each user is assigned a bitrate and Constraint 5e ensures that the user can decode its assigned bitrate by allocating enough RBs based on the user's spectral efficiency. Constraint 5f limits the percentage of RBs reserved for eMBMS at an eNB to α which is usually set to 60% [29]. It also guarantees that each eNB meets its unicast resource demand.

Time scale of optimization: In a cellular network, the network or user-state may vary or users may join or leave video streams, rendering a pre-computed network solution sub-optimal. To ensure a granular responsiveness to the dynamics, an algorithm must be able to solve the network reconfiguration problem in real-time, regardless of the number of users or state variations. The optimization model in Problem 1 is a quadratic program with Integer constraints and does not scale well with the number of users. For example, in the experiments conducted in Figure 6 with only 300 eMBMS users, it took 4 hours to solve the model once. Such a time-scale is not feasible for real-time environments where a solution maybe required every few seconds. Hence, we need a lightweight scalable algorithm that can find optimal or near-optimal network configurations in real-time.

IV. NIMBLE: Proposed Algorithm

The optimization problem solution defines the eMBMS network configuration including an SFN layout chosen for each video, the bitrates served in each cluster of that layout, and the physical layer parameters, i.e. the MCS and number of RBs for each transmitted video bitrate. In this section, we develop NIMBLE as a novel multi-stage heuristic-based resource management solution for eMBMS. Some of the components of the heuristics are inspired by the RTOP [7] design but upgraded to optimize overall user QoE rather than user bitrates only. Additionally, the design of NIMBLE integrates components to efficiently manage resources while

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3021229, IEEE Transactions on Multimedia

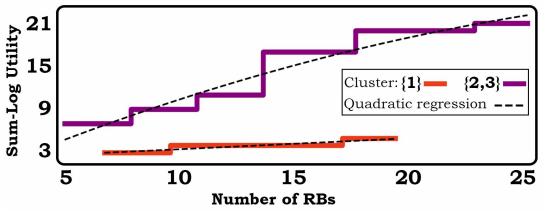


Fig. 4. A sample Utility vs RB graph with quadratic regression. *Lower and upper RB-bounds are minimum needed and total available RBs.*

accommodating the dynamics of user and network states in a cellular network.

A. NIMBLE Overview

The problem of selecting an optimal SFN layout for each video depends on the number of RBs allocated to each video and the consequent user grouping. This makes the problem combinatorial in nature and can result in exponentially-increasing outcomes. To reduce the computation time, we divide the overall process into four stages (Figure 3). The first stage calculates an approximate share of resources for each video (1a) and narrows down the choices of SFN layouts for videos to a set of candidate layouts (1b) with high utilities. The second stage then elects one SFN layout for each video. First, we generate utility vs resource graphs for the candidate layouts (2a); we then use these graphs to solve a simplified optimization problem for different candidate layout-combinations (2b) and; the combination with the highest utility is then identified.

The third stage first calculates the optimal share of RBs for all the videos in each SFN cluster of its chosen layout (3a) by solving an optimization problem. Then, for the given RBs, it calculates the optimal bitrates to transmit for each video in each cluster and the MCS and RB share for the bitrates (3b). Finally, stage 4 reconfigures the network if the utility of the chosen configuration exceeds the utility of the current configuration by a pre-defined margin (4a). NIMBLE then waits until the next reconfiguration instance and repeats the process if there are any active eMBMS sessions (4b). The rest of the section, describes these stages in further details.

B. Selecting Candidate SFN Layouts

In this section, we present the techniques introduced in NIMBLE to identify a set of candidate SFN layouts for each video. Our selection criteria involves calculating an approximate utility for all possible layouts (i.e. L) and choosing layouts with high utilities. As the optimal resource share per video is not known in advance, we consider an approximate RB distribution and choose multiple layouts as candidates to increase the probability of retaining the best layout. Note that the presented techniques are applied separately for each video as the resource distribution and user's channel condition may vary across videos, leading to a different optimal SFN layout.

Approximate Resource Share: We consider a proportional resource distribution among videos served by eMBMS. Let P_{vc} denote the approximate share of RBs for video v in cluster c:

$$P_{vc} = min(\rho_b \forall b \in c) \cdot \frac{|M_v|}{\sum_{v \in V} |M_v|}, \tag{6}$$

where ρ_b is defined in Equation 1. As an example, a cluster where the eNB with the least available RBs for eMBMS has 30 RBs available and that has to serve three eMBMS videos with 50, 100 and 150 users respectively, will assume $30 * \frac{50}{50+100+150} = 5$ RBs allocated to the first video.

We calculate the approximate RB share for individual clusters in every possible layout. Note that, to improve the accuracy of utility estimation, the distribution should be chosen according to the fairness-metric of the system (which in this case is PF).

Multiple Candidate Layouts per Video: The utility estimate based on the approximate RB share may be sub-optimal and the optimal RB distribution can only be identified when the whole configuration is determined. Therefore, to increase the probability of retaining the best SFN layout for a video when applying this heuristic, we choose multiple layouts as our candidates for each video. We achieve this by classifying the layouts based on the number of clusters in them and, for each class picking a candidate layout as the one with the highest utility in each class (Figure 5). The utility of a layout l_v is calculated by finding the best *user-groups* for each $c \in l_v$, assuming P_{vc} as the available RBs, as explained below.

Choosing The Best User-Grouping: Depending on P_{vc}, users associated with the SFN cluster c can be split into multiple *user-groups* with each group served a different video bitrate. Serving too many bitrates reduces the share of RBs per *user-group*, and may reduce the achievable utility. In [7], we conducted extensive experiments and showed that 90% of the time, the optimal number of *user-groups* is no more than two. So for our heuristics, we limit the maximum number of *user-groups* per video served in an SFN cluster to two.

We restrict the MCS assigned to a *user-group* to that of the user with the worst channel condition. This avoids decoding errors or frame losses for any user, but an inefficient user placement may unnecessarily limit the spectral efficiency of a *user-group* and hence the achievable bitrates. Therefore, we calculate the utility that different user-grouping combinations can achieve as explained below and choose the grouping that yields the highest utility.

Identifying *user-groups* involves partitioning users into groups and determining the bitrates served to each *user-group*. Our objective is to identify the user-grouping in each SFN cluster that maximizes the total utility at the decision instance. Exhaustively testing all possible combinations of *user-groups* and bitrates for each video takes $\mathcal{O}(|R_v| \cdot |M_v|^{|M_v|})$ to solve. This approach would not scale for a large number of users.

In NIMBLE, we first calculate the bitrate component of the utility using the RTOP user-grouping algorithm [7]. This algorithm pre-groups users based on their Channel Quality Indicator (CQI) values and finds the highest bitrate that can be assigned to each *CQI-group* with the available RBs (P_{vc}).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3021229, IEEE Transactions on Multimedia

Although this algorithm does not measure the switching-penalty that each user may incur, it runs in $O(|R_v|^2)$ to find the maximum bitrate-utility and the best user-grouping of a video in an SFN cluster for a given number of RBs. As the number of streamed bitrates ($|R_v|$) is usually small for live video streaming [32], this algorithm allows real-time computation of the problem.

Utility with Switching Penalties: We run the user-grouping algorithm for all clusters ($c \in l_v$) to identify the user grouping that achieves the highest utility and find the bitrates (r_m) that each user ($m \in M_v$) will receive. We then calculate the switching component of the utility for each user using Equation 4. NIMBLE determines the current streaming bitrate r'_m as presented in Section III-B and calculates switching utility accordingly.

The total utility of layout l can then be determined by adding both bitrate and switching components for all users. This process is repeated for each video for every $l \in L$ to estimate the approximate achievable utilities. The layouts with highest utility in each of the N layout-class (Figure 5) are chosen to form a set of *candidate* SFN layouts per video.

C. Electing an SFN Layout for Each Video

Identifying the optimal network configuration implies determining both the optimal layouts and the optimal resource shares for all eMBMS videos. The approximate resource distribution (P_{vc}) used in the previous stage may yield sub-optimal results, as it overlooks the channel condition of users. As a video may need additional resources to stream the lowest bitrate to users with weak channels, the approximation may also lead to infeasible solutions.

In this step, we optimize these eMBMS configuration decisions over various combinations of the *candidate* layouts for every video while considering all possible RB distributions among eMBMS videos. Note that changing the RB share of a video would impact the achievable utility for a given SFN layout as it could impact user-grouping decisions. The optimization also considers both network and user constraints defined in Section III-B.

Utility vs RB graphs: For each candidate layout l_v, we calculate the bitrate utility of every underlying cluster for every possible RB share using the user-grouping algorithm and represent it as a graph function, denoted by G. Figure 4 illustrates the function for a layout that has two clusters. Note that the two clusters could have different resource availability, e.g., due to different cell load in different clusters. Due to the discrete nature of video bitrates and allocated MCS, the bitrate-utility represents a stair function in RB.

The utility of layout l_v is calculated as the aggregate utilities from all its underlying clusters. If X_{vc} represents the RBs allocated to v in each cluster $c \in l_v$ then, for a given set of X_{vc}, the utility of l_v can be calculated as:

$$h(l_v) = \sum_{c \in l_v} G(X_{vc}) \tag{7}$$

Equation 7 represents the objective function used for selecting the optimal layouts. To speed up this selection, we

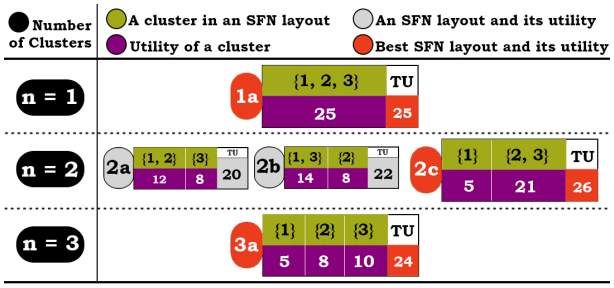


Fig. 5. Example of SFN layouts classified by number of clusters in them. *A candidate layout is chosen in each class.*

apply quadratic regression to the discrete graph function h to obtain a continuous function, h', as illustrated by dashed lines in Figure 4. The quadratic regression provided sufficiently accurate and fast solution in comparison to the less accurate linear regression and the slower higher-degree polynomials.

$$h'(l_v) = \sum_{c \in l_v} i_{vc} \cdot X_{vc}^2 + j_{vc} \cdot X_{vc} + k_{vc}, \tag{8}$$

where i_{vc}, j_{vc} and k_{vc} are the quadratic regression coefficients. The continuous function provides faster computation by allowing us to define a linear program, instead of an integer linear program that the discrete graphs G (Equation 7) would produce.

Unlike the *CQI-groups* used for bitrate-factor, switching-factor involves analyzing each user's state individually and hence can take longer computation time. Hence we defer utility-adjustment for switching-factor until a later stage, as at this stage we have to calculate the function values for multiple possible RB shares for each cluster in every candidate SFN layout and repeat the process for each video.

Selecting Optimal Layout Combination: In this step, we determine the optimal SFN layout l_v for each video by maximizing the regression-based bitrate-utility function (Equation 8) while satisfying resource and QoE constraints.

Problem 2: *Maximizing utility for a combination of layouts*

$$max \sum_{v \in V} h'(l_v) \tag{9a}$$

subject to

$$\sum_{v \in V} \sum_{c \in l_v} \beta_{blc} \cdot X_{vc} \leq \rho_b, \forall\, b \in B \tag{9b}$$

$$X_{vc} \geq D_{vc}, \forall\, v \in V, c \in l_v, \tag{9c}$$

where D_{vc} is the lower bound for X_{vc} and represents the minimum number of RBs needed to attain the lowest bitrate for all users of v in c.

Constraint 9b is similar to Constraint 5f and limits the RBs available to eMBMS users in a cluster. Constraint 9c ensures that each video gets enough RBs to at least attain the lowest bitrate for all users and avoid stalls.

Solving Problem 2 for a particular combination of candidate layouts per video provides us with the resource shares for each video based on quadratic regression. We use this information to find the bitrate that each user would receive and calculate the

switching-factor of the utility to obtain the total approximate utility achievable by the combination. Repeating the process for various possible combinations of candidate layouts, gives us the combination with the highest achievable utility and optimal $l_v \forall v \in V$.

D. Optimal Network Configuration

To acquire the complete network configuration, we solve Problem 2 one more time for the chosen SFN layout combination while using the discrete utility function h (Equation 7) instead of the quadratic regression. The problem solution determines the optimal X_{vc} for each video and cluster. These values are then input to user grouping algorithm and the techniques explained in Section IV-B are utilized to determine the bitrates to transmit for each video and the associated MCS values and number of RBs.

In our experiments (Section V), NIMBLE solves the network configuration problem in less than 500ms on a dual-core i7 processor laptop with 16GB RAM. This computation time is well within the expected time constraints for a cellular network [33]. Hence, we argue that NIMBLE is fast enough to mange eMBMS in typical service-area sizes regardless of the number of eMBMS users. In the following section we further present a few additional practical concerns for real-world deployments of eMBMS in dynamic networks.

E. Controlling Network Reconfiguration

In a dynamic network, the user and network state changes continuously. Hence, eMBMS network configuration should be updated using NIMBLE to maintain optimized operation. Changing the SFN layout or bitrates to serve for any video implies that the MCE, based on NIMBLE solution, should update the multicast control channel (MCCH) and multicast transport channel (MTCH). The information carried by MCCH includes MCS and sub-frame allocation. Changes in scheduling information are sent to all the involved eNBs and eNBs advertise MCCH to users [4].

Frequent reconfigurations increase the overhead cost associated with these actions and message exchanges. Albeit minimal, frequent layout reconfigurations may also increase the number of transport blocks lost in the network when altering user-cluster association due to the lack of re-transmission mechanism in eMBMS [4]. To minimize the impact of network reconfigurations, we introduce two *control* parameters to NIMBLE.

Utility Increment Threshold (δ): Based on the amount of variability in the system, the new network configuration may have insignificant utility improvement in comparison to the current one. We define a threshold ratio δ and only reconfigure the network when the increase in utility exceeds the threshold and both configurations are valid. If the current network configuration becomes infeasible, e.g.,cannot deliver stall-free video to all the users anymore, the new configuration is applied regardless of the amount of gain in utility.

Reconfiguration Trigger (τ): In highly dynamic network scenarios with frequent changes in user and network states, periodic execution represents a good approach to trigger NIMBLE. Setting the trigger period, denoted by τ, too low can result in frequent network reconfigurations while high τ values can slow the responsiveness to the changes in network or user states. An operator can set τ to reflect the characteristics of their network.

In fairly-static network scenarios, an event-based approach can be used to decide when to re-run NIMBLE. We can count the number of events such as users leaving or joining eMBMS session, beginning or end of an eMBMS session, users' channel condition changed beyond certain limits etc. A NIMBLE re-run can be triggered when an operator-defined threshold is met.

Determining the best trigger design involves defining various parameters such as, what constitutes as highly-dynamic or fairly-static network and importance or optimal values for the aforementioned events. Such definitions are network and operator specific and beyond the scope of this work. In our experiments, we choose periodic reconfiguration trigger for NIMBLE and analyze the impact of different interval values on our performance metrics in the given network scenarios (Section V-B1).

Algorithm 1 Complete NIMBLE algorithm

Input: M_{vlc}: Users of v in cluster $c \in l$; C_N: Current network config. and U_N: its utility. See Table I for other inputs.
Output: SFN cluster and user-groups for each video

while $Active\_eMBMS\_Sessions$ **do**
  for $v \in V$ **do**
    for $l \in L$ **do**
      $utility, Graph[v,l] \leftarrow$ **LAYOUT_U**(v,l)
      if $utility > maxU[v,|l|]$ **then**
        $maxU[v,l] \leftarrow utility; Candidates[v,|l|] \leftarrow l$
  $Cartesian \leftarrow \{(l_1..l_v) \mid l_v \in Candidates[v]$ **for** $v \in V\}$
  $U_{max} \leftarrow 0$
  for $combo \in Cartesian$ **do** ▷ *Layout combinations*
    Get Utility U by solving Problem 2
    if $U > U_{max}$ **then**
      $U_{max} \leftarrow U; Optimal\_Combo \leftarrow combo$
  if $C_N \notin Cartesian \vee U_N < \delta \cdot U_{max}$ **then**
    Solve Prob. 2 for $Optimal\_Combo$ without regression to get optimal RB shares and user-groups for each video
  $sleep(\tau)$ ▷ *minus the time consumed in steps above*

LAYOUT_U(v,l) ▷ *Utility of an SFN layout*
$PF\_utility \leftarrow 0; Graph \leftarrow []$
for $c \in l$ **do**
  $maxRBs \leftarrow$ *min*$(\rho_b$ **for** $b \in c)$ ▷ *Max. available RBs*
  for $RBs \leftarrow 1$ *to* $maxRBs$ **do**
    $Graph[RBs] \leftarrow$ **User_Grouping**(M_{vlc}, RBs)
  $PF\_RBs \leftarrow \frac{|M_v|}{sum(M_w \forall w \in V)} * maxRBs$
  $PF\_utility += Graph[PF\_RBs]$
return $PF\_utility, Graph$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3021229, IEEE Transactions on Multimedia

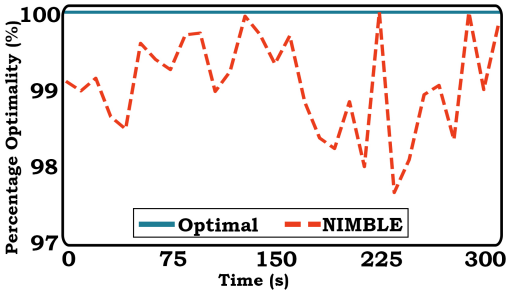


Fig. 6. Percentage optimality of the NIMBLE algorithm

F. Complete NIMBLE Algorithm

Algorithm 1 presents all our steps based on the aforementioned techniques. Before running the algorithm, we perform a sanity check to remove infeasible layouts for each video, i.e., layouts that are incapable of serving at least the minimum bitrate to all video users.

We start the algorithm by using the layout-utility function (Lines 16-22) to find approximate utility achievable by different SFN layouts for each video (Lines 2-4). We then classify layouts based on the size of the layout, i.e. the number of clusters in them and from each class, choose the layout with the highest utility as a candidate (Lines 5-6). Then, we explore all possible combinations of candidate layouts for different videos (Line 7) and find the combination with the highest sum-utility (Lines 8-12) by solving Problem 2 with regression-based utility (Equation 8).

Then, we solve Problem 2 (Line 14) without regression (Equation 7) to find the optimal resource shares in each cluster of the chosen video layouts and run user-grouping algorithm to find the optimal *user-groups*, bitrates and physical-layer parameters in each cluster.

Finally, if the utility of the best combination is higher than the current configuration's utility by a factor of δ or if the current configuration is not feasible anymore (Line 13), we apply the new network configuration. We wait until it is time to run the algorithm again (Line 15), and repeat the process until there are no active eMBMS sessions (Line 1).

G. Optimality of NIMBLE

If there is only one video served through eMBMS in the service area then the solution computed by NIMBLE is always optimal. This is because all the RBs available for eMBMS can be used by the single video and the heuristics in Step 2 of Figure 3 is not applied. However, due to the heuristic nature of NIMBLE, when there are more than one videos served, the computed solution may not be globally optimal.

To establish the optimality of NIMBLE, we compare the utility achieved by NIMBLE with the optimization model in Section III for three eMBMS videos. We conduct experiments using the same simulation parameters and variable values as in Section V-A. However, due to the time complexity and high computation time of the optimization model, the total number of eMBMS users is reduced to 300 in these experiments.

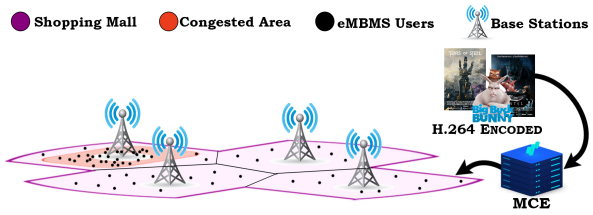


Fig. 7. Simulation Scenario: *Shopping Mall with 4 eNBs (1 congested)*

Figure 6 shows that the average gap between the optimal network configurations and NIMBLE is 1% and the utility achieved by NIMBLE stays within 2.5% of the globally optimal solution at all times. This shows that even in a dynamic network with multiple variables, NIMBLE produces highly accurate results. Coupled with its scalability and computation-efficiency, this enables NIMBLE to be deployed in practical scenarios and real-world networks.

V. Performance Evaluation

Our evaluation is conducted using a trace-driven simulation testbed. In this section, we first present our evaluation setup. Then we evaluate the performance of NIMBLE in typical eMBMS scenarios and analyze the impact of its *control* parameters. Finally, we compare the performance of NIMBLE to state-of-the-art approaches.

A. Evaluation Setup

eMBMS can address resource allocation challenges in densely populated or crowded areas. For realistic evaluation of NIMBLE, we simulate a practical use case of eMBMS. We consider an eMBMS service area covering a large shopping mall (Figure 7) with four eNBs arranged in a hexagonal grid. The mall consists of an entertainment zone with restaurants, theater, resting area, food court etc. Due to the lack of real traces available in literature for shopping mall user-patterns, we define three different types of users in the mall to capture various patterns and achieve relatively realistic scenarios.

Furthermore, to simulate a congestion scenario, we assume that 30% of the eMBMS users are located in the entertainment zone and that the closest eNB is congested. The rest of the eMBMS users are distributed uniformly over the service area. The three types of user classes based on mobility patterns are defined as follows:

Resting: These are static users and during the duration of our simulation are assumed to not change their location. 80% of the resting users are located in the entertainment zone while the rest are distributed uniformly across the mall.

Shopping: These users moves around the mall using Random Way Point [34] (RWP) mobility model. To emulate the time spent inside the shops, we set a high pause time (Table II) for this category.

Browsing: These users do not spend much time at one shop. We also use RWP model with a small range for pause time (Table II) to mimic the behavior of browsing users. These users exhibit the highest mobility in our simulation scenario.

We generate 900 mobility traces with equal probability of each user type. Each trace defines the initial location and the mobility pattern of the user for the duration of the simulation. We assign these traces to our 1200 eMBMS users by allowing every user to randomly choose one of the traces as the simulation starts. Hence, as multiple users choose the same trace, they form a group of two or more users moving together (e.g., a family or a group of friends). We consider Multi-Path Fading AWGN channel model [35] for device-eNB links. The parameters of this model and achievable spectral efficiencies are listed in Table II [29].

Each eMBMS user reports its SINR from each cluster, which is used to calculate the achievable MCS based on the BLER (Block Error Rate) vs SINR curves [35]. The target BLER is typically set to 1% in eMBMS [4], contrary to 10% in unicast, as there are no physical layer re-transmissions. Additionally, the device averages its SINR over a duration of 250ms for each SFN cluster to choose the best cluster and applies handover-hysteresis of $1dB$ to avoid ping-pong effect.

Each of the 1200 eMBMS users streams one of three live videos (400 users per video). The raw videos are obtained from [36] and five minutes of every video are encoded at three different bitrates using H.264 at 24 frames/sec (fps), in the order IPBBBPBB i.e., 8 frames per group of picture (GOP). The bitrates include low (400kbps), medium (1.5Mbps) and high (4Mbps) video qualities. A 10% Forward Error Correction (FEC) is also assumed to be encoded in each frame, and a frame is considered lost if more than 10% of its transmitted bytes are lost. Note that we consider transmission errors represent the main source of frame losses. We assume that eMBMS traffic would not experience random queuing delays as it is transmitted over preserved resources over the wireless interface. We also assume that eMBMS traffic would also be well-provisioned over the wired part of the network (e.g., backhaul and fronthaul). The encoded videos are used to generate trace files that includes individual frame sizes. When simulating frame-transmission on the physical-layer, depending on the frame size, each frame is split into one or more transport blocks. Each eMBMS client subscribes to the highest available video quality decodable given the user SINR. When switching between different qualities, the client waits until an end of a GOP, to ensure a smooth switching experience.

Our key performance metrics are

- Throughput: Average size of decodable frames received by users per second.
- Transport blocks (TB) and frames lost in the network.
- Total number of bitrate switches encountered by users over the duration of the simulation.
- Stall duration: We define stall as more than one consecutive frames not played by a user, either due to network loss or a missing dependency.
- QoE: We use the regression-based QoE model [25] defined as: $-56.6P_r + 0.007\overline{B} + 0.0007\overline{B_s} + 54.0$, where P_r is the stall percentage, $\bar{B}$ is the average bitrate and $\bar{B_s}$ is the average bitrate switch magnitude.

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Cellular Layout	Hexagonal grid with 4 eNBs
eMBMS Service Area	1250 m x 875 m
eNB Tx Power	10 Watts
Frequency, bandwidth, RBs	2.1 GHz, 20 MHz and 100
Path Loss Model	Log-Normal Shadowing
Path Loss Exponent	4.5 (Indoor Obstructed)
Noise Density & UE Noise	-174 dBm/Hz & 7 dB
Channel Model	Multi-path Fading AWGN [35]
Handover Hysteresis Margin	$1dB$ of average-SINR (over 250ms)
Spectral Efficiency (bits/RB) from CQIs 1 to 15 [4]	[20, 31, 50, 79, 116, 155, 195, 253, 318, 360, 439, 515, 597, 675, 733]
User Mobility Model	Random-Way Point [34]
User Movement Speed	Walking: 1∼1.5 m/s-Stationary:0m/s
Pause/Waiting Times	Shopping: 0∼300s-Browsing: 0∼30s

- Fairness: We measure f-index [28] defined as, $1 - \frac{2\sigma}{H-L}$, where σ is the standard deviation of user QoE scores, H is the upper QoE bound (100) and L is the lower QoE bound (0).
- Network reconfiguration rate: Changes made to SFN layout or streaming bitrates of a video per minute.

B. Analysis of control parameters

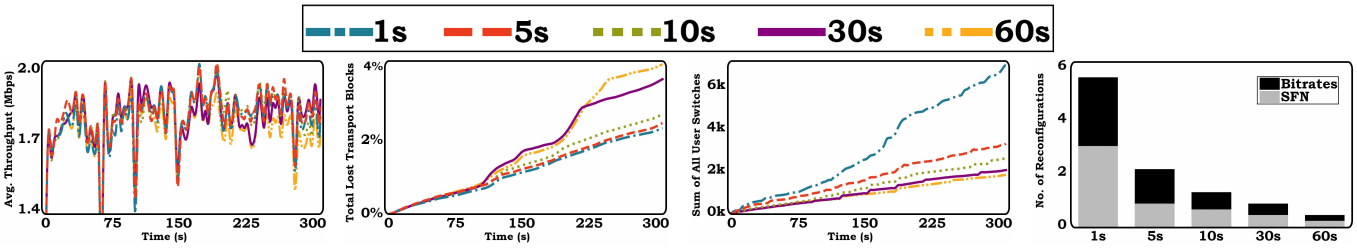
In this section, we analyze the impact of our two *control* parameters (Section IV-E). We vary the parameter values and examine the impact on average user throughput, total TBs lost in the network, total bitrate switches experienced by all the users and the network reconfiguration rate. For each scenario, we repeat each experiment 5 times by varying user mobility traces and present the average results.

1) Tuning periodic trigger interval: For this set of experiments, we fix the increment threshold (δ) to 2% and vary the periodic reconfiguration trigger interval between 1s and 60s. Delaying network reconfiguration (e.g. $i = 60s$) resulted in lesser reconfiguration overhead (Figure 8d) but also reduced NIMBLE's responsiveness to changes in network or user condition and incurred high ($\approx 4\%$) network losses (Figure 8b).

On the other hand, configuring network too often (every 1s) increased the switches in bitrates (Figure 8c) and overhead costs but reduced losses by responding quickly to the system dynamics. Due to user mobility and δ being same for these experiments, the throughput trend was similar for different recalculation intervals. However, due to fewer losses in network and quicker response to changes in user channel condition, approaches that ran more often achieved higher throughput (Figure 8a). In comparison to $i = 60s$, $i = 5s$ increased the average system throughput by 4%.

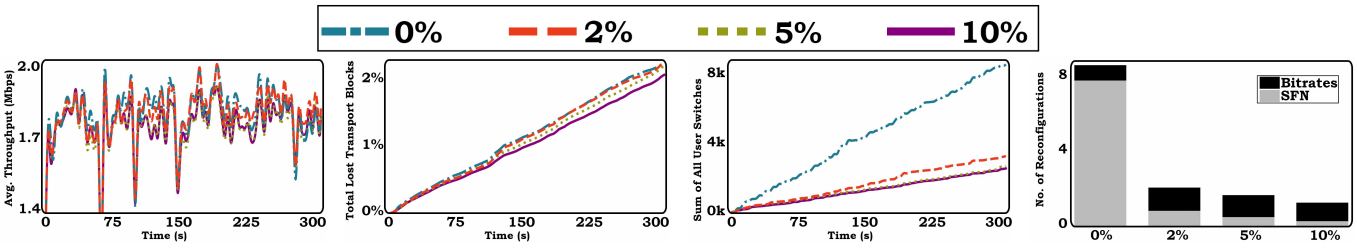
These experiments demonstrate the importance of selecting an appropriate reconfiguration interval and show that prolonging reconfiguration can reduce the responsiveness to users channel condition and deteriorate user experience by increasing the losses or reducing the throughput, whereas reconfiguring too often can negatively impact user experience by increasing the number of bitrate switches. For the following

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3021229, IEEE Transactions on Multimedia



a *Avg. Throughput Per User* b *Total TBs Lost in Network* c *Total Bitrate Switches in Network* d *Reconfigurations per Minute*

Fig. 8. Analyzing Recalculation Interval i at $\delta = 2\%$: *Delaying reconfiguration may reduce switches but results in higher losses.*



a *Avg. Throughput Per User* b *Total TBs Lost in Network* c *Total Bitrate Switches in Network* d *Reconfigurations per Minute*

Fig. 9. Analyzing Increment Threshold δ at $i = 5s$: *Lower values can achieve higher throughput but incurs higher switches and reconfiguration overhead.*

experiments, we choose $i = 5s$ which, in our simulation scenario, keeps bitrate switches and reconfiguration overhead to a reasonable value while maintaining responsiveness to network-state and user channel condition.

2) Tuning utility increment threshold (δ): For this set of experiments, we consider a fixed periodic interval i of 5 seconds and vary δ between 0% and 10%. Higher δ values decrease the probability of reconfiguring network based on bitrate gains in an effort to reduce bitrate switches. This trend can be seen in Figure 9 where $\delta = 0\%$ achieved the highest throughput (Figure 9a) but also the highest number of switches (Figure 9c).

As mentioned in Section IV-E, regardless of the value of δ, NIMBLE reconfigures the network if the current configuration has become infeasible, i.e. can cause losses for any user. Therefore, the difference in percentage of lost TBs was insignificant across different values of δ, as even though setting δ higher reconfigured network less often (Figure 9d), it did not incur higher losses (Figure 9b), by ensuring that infeasible solutions are always avoided.

As NIMBLE ran every 5 seconds and found a solution with BLER$\leq$ 1% for each user, the total percentage of lost TBs was around 2%. The additional 1% is accounted by the possible degradation in channel condition of some users during the 5-second interval due to the path loss or mobility pattern. Based on these experiments and analysis, we pick $\delta = 2\%$ for NIMBLE in the following experiments, which avoids network reconfiguration on slight utility-increments, while maintaining responsiveness to user experience through a 3% switches-reduction in comparison to $\delta = 0\%$ and a 2.5% throughput-increase in comparison to $\delta = 5\%$ and $\delta = 10\%$.

C. Comparison with State-of-the-art Algorithms

In this section, we compare the performance of NIMBLE to the following approaches:

BoLTE [6]: Creates SFN clusters to maximize PF-utility but does not consider grouping users based on their channel condition. The proposed algorithm assumes a single bitrate per video. For a fair comparison, the utility of an SFN layout is calculated by assigning the best achievable bitrate for each video by the heuristics proposed in BoLTE.

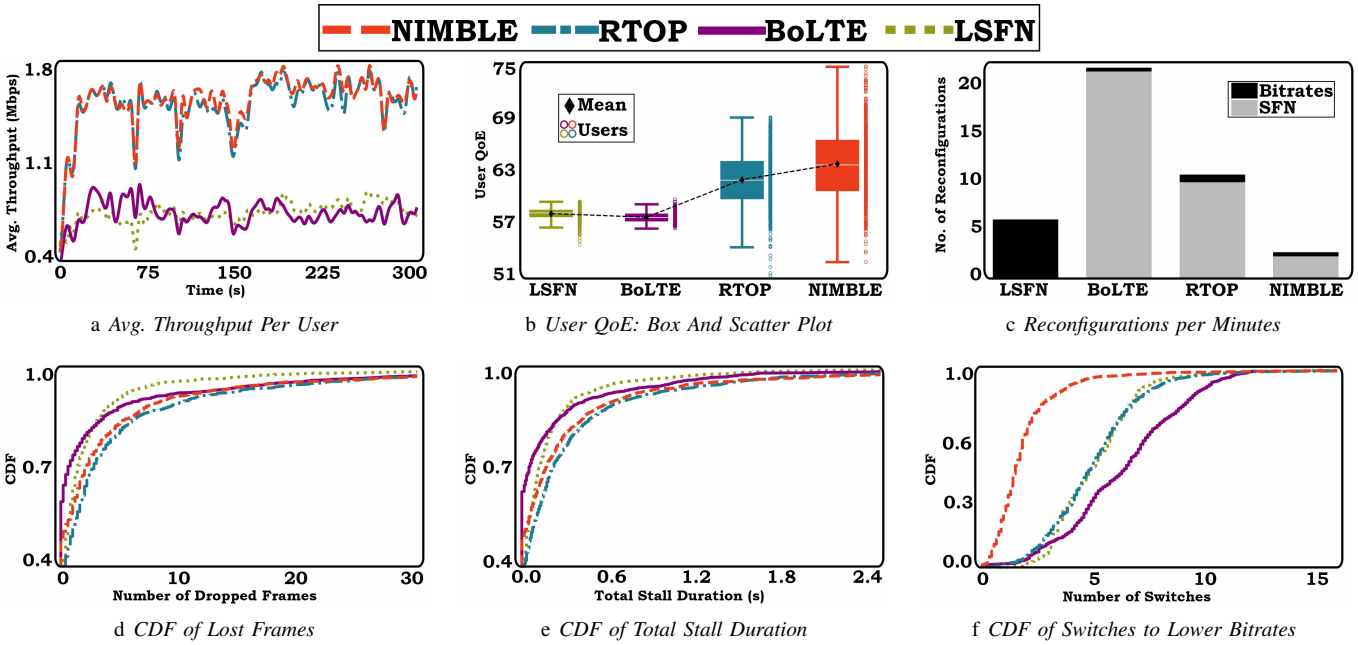
One Large SFN (LSFN): In [5], the authors propose partitioning users into groups based on their channel condition, but assume that all eNBs in the service area are always synchronized, i.e., creating one large SFN cluster. The proposed scheme only solves the problem for one eMBMS video. We extend [5] based on our optimization model to work with multiple videos and call the approach LSFN.

RTOP [7]: This algorithm solves a joint optimization problem for creating SFN clusters as well as *user-groups* to maximize a bitrate-based PF utility. RTOP does not consider the variation in network or user state over time or the impact of bitrate-switches on user QoE.

For a fair comparison, we run each algorithm every 5 seconds for PF-utility maximization over the same network and user-state. In this comparison, we also repeat the experiment 5 times by varying user mobility traces and report the average metric results.

1) System Throughput: Figure 10a shows the average system throughput. As LSFN does not consider SFN clustering, the RBs available to eMBMS were restricted by the congested eNB and all the users had to be served with limited RBs. On the other hand, BoLTE lacks user-grouping and could not assign rates to users commensurate to their channel conditions. NIMBLE and RTOP considered both these factors and hence increased the average throughput by up to 150%.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2020.3021229, IEEE Transactions on Multimedia



a *Avg. Throughput Per User* b *User QoE: Box And Scatter Plot* c *Reconfigurations per Minutes*

d *CDF of Lost Frames* e *CDF of Total Stall Duration* f *CDF of Switches to Lower Bitrates*

Fig. 10. Results of comparing algorithms: *NIMBLE reduces network reconfigurations and serves users with higher throughput and fewer switches.*

NIMBLE achieved slightly higher throughput than RTOP. This is because RTOP incurs slightly higher drops on the users (Figure 10d) due to frequent network reconfigurations (Figure 10c). By using the *control* parameter δ, NIMBLE stabilizes the network and user state, and increases the successfully received frames and hence the average throughput.

2) Lost Frames and Stalls: Our implementation of each algorithm runs every 5 seconds and has an explicit or implicit constraint to avoid frame losses or stalls. Hence all the algorithms reacted well to the possible variation in user channel condition and mobility. Almost all the users lost less than 30 frames (0.4%) and stalled for less than 3 seconds (1%) during the five-minutes of simulation (Figure 10d and 10e).

3) Switches in Bitrates: Based on subjective evaluation, less than one switch per minute is acceptable by users and does not annoy them [37]. This implies 5 switches per our 5-minute simulation duration which was achieved by only 40% users with BoLTE and around 55% with LSFN or RTOP. However, with NIMBLE around 98% of users encountered less than 5 switches (Figure 10f). NIMBLE achieved lower switch count while still serving users with the highest throughput (Figure 10a). This is because unlike other algorithms, NIMBLE takes into account the impact of switching bitrates when reconfiguring the network and avoids too many switches which can annoy users and have a negative effect on video QoE.

4) QoE and Fairness: With least number of switches and highest throughput, NIMBLE provided the best QoE for users, followed by RTOP, which achieved similar throughput but switched bitrates more often than NIMBLE. BoLTE and LSFN focus on static networks, ignore the impact of switches and also fail to maximize the throughput as they do not consider combined advantage of SFN clustering and user-grouping, hence had lower QoE values. In comparison to LSFN, NIMBLE increased the average user QoE by 13% (Figure 10b).

Considering f-index fairness (Table III), LSFN and BoLTE performed slightly better. This result is because most of the users had *equally bad* experience. Hence, the system seems fair but actually performs worse because of the inability to react to the underlying network or user state. On the contrary, NIMBLE assigned bitrates to users commensurate to their channel condition and available RBs across eNBs.

To further analyze this, we look at the number of users that achieved a higher (or lower) QoE with other algorithms in comparison to NIMBLE. As shown in Table III (Row 2), NIMBLE improved QoE for more than 1000 users at the expense of only 54 users for LSFN, 30 users for BoLTE and 135 users for RTOP. Also, as shown in Figure 10b, the decrease in QoE is marginal for most of these users.

5) Network reconfiguration rate: NIMBLE incurred less signaling overhead (Figure 10c) by reconfiguring the network less often, especially in comparison to BoLTE and RTOP. While the actual acceptable overhead cost depends on operators and their network preferences, these results show that with efficient network management and configuration, NIMBLE was able to achieve higher user QoE with the least reconfigurations. This reduction is due to the *control* parameters (Section IV-E) considered by NIMBLE when reconfiguring the network.

VI. Conclusion

eMBMS promises improved resource utilization and user perceived QoE. In this paper, we formulated an optimization model and proposed a heuristics-based algorithm to maximize the end-user QoE by optimally configuring the eMBMS service area. NIMBLE design integrates control parameters to accommodate the dynamic nature of cellular networks, user mo-

TABLE III
FAIRNESS ANALYSIS

Algorithm	LSFN	BoLTE	RTOP	NIMBLE
f-index	0.96	0.97	0.86	0.83
Rel. QoE	1116 ↓ 54 ↑	1154 ↓ 30 ↑	1064 ↓ 135 ↑	-

bility and varying channel condition over time and minimize its impact on user QoE. Using extensive evaluation, we showed how these parameters can be tuned for a better trade-off between network responsiveness, efficiency and stability. When compared with state-of-the-art approaches we demonstrated that NIMBLE improves user QoE and various underlying metrics. This improvement was achieved because NIMBLE jointly optimizes SFN clustering and user grouping problems when reconfiguring the network and allocating resources to different eMBMS video sessions. NIMBLE achieves 150% increase in user throughput, 75% reduction in user switches, 15% increase in average user QoE, all while reducing the network reconfiguration count by 90%.

ACKNOWLEDGMENT

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 13/IA/1892, and is co-funded under the European Regional Development Fund under Grant number 13/RC/2077.

REFERENCES

[1] *Twitch*, Retrieved 04-July-2020 from https://twitchtracker.com/.
[2] Ericsson, *Ericsson Mobility Report*, June 2019.
[3] Cisco, *Annual Internet Report, 2018-2023 (White Paper)*, March 2020.
[4] 3GPP, "Multimedia Broadcast/Multicast Service (MBMS); Architecture and functional description," 3GPP, Tech. Rep. 23.246, 2015. [Online]. Available: www.3gpp.org/DynaReport/23246.htm
[5] J. Chen, M. Chiang, J. Erman, G. Li, K. Ramakrishnan, and R. Sinha, "Fair and optimal resource allocation for LTE multicast (eMBMS): Group partitioning and dynamics," *IEEE Conference on Computer Communications (INFOCOM)*, October 2015.
[6] R. Sivaraj, M. Arslan, K. Sundaresan, S. Rangarajan, and P. Mohapatra, "BoLTE: Efficient network-wide LTE broadcasting," *IEEE 25th International Conference on Network Protocols (ICNP)*, April 2017.
[7] A. Khalid, A. Zahran, and C. Sreenan, "RTOP: Optimal User Grouping and SFN Clustering for Multiple eMBMS Video Sessions," *IEEE Conference on Computer Communications (INFOCOM)*, June 2019.
[8] 3GPP, "Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs," 3GPP, Tech. Rep. 26.346, 2015. [Online]. Available: www.3gpp.org/DynaReport/26346.htm
[9] D. Barquero, W. Li, M. Fuentes, J. Xiong, G. Araniti, C. Akamine, and J. Wang, "5G for Broadband Multimedia Systems and Broadcasting," *IEEE Transactions on Broadcasting*, vol. 65, no. 2, pp. 351–355, 2019.
[10] A. Khalid, A. Zahran, and C. Sreenan, "An SDN-based Device-aware Live Video Service for Inter-domain Adaptive Bitrate Streaming," *10th ACM Multimedia Systems Conference (MMSys)*, pp. 121–132, 2019.
[11] M. Hosseini, D. T. Ahmed, S. Shirmohammadi, and N. D. Georganas, "A Survey of Application-Layer Multicast Protocols," *IEEE Communications Surveys & Tutorials*, vol. 9, September 2007.
[12] J. Wu, B. Cheng, and M. Wang, "Improving Multipath Video Transmission With Raptor Codes in Heterogeneous Wireless Networks," *IEEE Transactions on Multimedia*, vol. 20, no. 2, pp. 457–472, 2018.
[13] J. Wu, B. Cheng, M. Wang, and J. Chen, "Energy-Efficient Bandwidth Aggregation for Delay-Constrained Video Over Heterogeneous Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 1, pp. 30–49, 2017.
[14] J. Wu, C. Yuen, B. Cheng, M. Wang, and J. Chen, "Streaming High-Quality Mobile Video with Multipath TCP in Heterogeneous Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 9, pp. 2345–2361, 2016.
[15] F. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, no. 1, pp. 33–37, Jan 1997.
[16] F. Hou, L. Cai, P. Ho, X. Shen, and J. Zhang, "A cooperative multicast scheduling scheme for multimedia services in IEEE 802.16 networks," *IEEE Trans. on Wireless Communications*, vol. 8, no. 3, March 2009.
[17] J. Yoon, H. Zhang, S. Banerjee, and S. Rangarajan, "Muvi: A multicast video delivery scheme for 4G cellular networks," *ACM MobiCom*, 2012.
[18] H. Won, H. Cai, D. Eun, K. Guo, A. Netravali, I. Rhee, and K. Sabnani, "Multicast scheduling in cellular data networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 9, September 2009.
[19] O. Eltobgy, O. Arafa, and M. Hefeeda, "Mobile Streaming of Live 360-Degree Videos," *IEEE Transactions on Multimedia*, 2020.
[20] S. Almowuena, M. Rahman, C. Hsu, A. Hassan, and M. Hefeeda, "Energy-Aware and Bandwidth-Efficient Hybrid Video Streaming Over Mobile Networks," *IEEE Transactions on Multimedia*, vol. 18, no. 1, pp. 102–115, 2015.
[21] A. Argyriou, D. Kosmanos, and L. Tassiulas, "Joint time-domain resource partitioning, rate allocation, and video quality adaptation in heterogeneous cellular networks," *IEEE Transactions on Multimedia*, vol. 17, no. 5, pp. 736–745, 2015.
[22] ITU, "Vocabulary for performance, quality of service and quality of experience," ITU, Tech. Rep. P.10/G.100, 2017.
[23] H. Chang, C. Hsu, T. Hofeld, and K. Chen, "Active Learning for Crowdsourced QoE Modeling," *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp. 3337–3352, 2018.
[24] Y. Liu, S. Dey, F. Ulupinar, M. Luby, and Y. Mao, "Deriving and Validating User Experience Model for DASH Video Streaming," *IEEE Transactions on Broadcasting*, vol. 61, no. 4, pp. 651–665, Dec 2015.
[25] Z. Duanmu, A. Rehman, and Z. Wang, "A Quality-of-Experience Database for Adaptive Video Streaming," *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 474–487, June 2018.
[26] W. Song and D. Tjondronegoro, "Acceptability-Based QoE Models for Mobile Video," *IEEE Transactions on Multimedia*, vol. 16, no. 3, pp. 738–750, 2014.
[27] R. Jain, A. Durresi, and G. Babic, "Throughput fairness index: An explanation," in *ATM Forum contribution*, vol. 99, no. 45, 1999.
[28] T. Hoßfeld, L. Skorin-Kapov, P. E. Heegaard, and M. Varela, "Definition of QoE Fairness in Shared Systems," *IEEE Communications Letters*, vol. 21, no. 1, pp. 184–187, Jan 2017.
[29] EBU, "Delivery of broadcast content over LTE networks," European Broadcasting Union, Tech. Rep. TR 027, 2014. [Online]. Available: https://tech.ebu.ch/docs/techreports/tr027.pdf
[30] A. Zahran, J. Quinlan, K. Ramakrishnan, and C. Sreenan, "SAP: Stall-Aware Pacing for Improved DASH Video Experience in Cellular Networks," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, 2017, pp. 13–26.
[31] R. Mok, E. Chan, and R. Chang, "Measuring the quality of experience of HTTP video streaming." in *Integrated Network Management*, 2011, pp. 485–492.
[32] K. Pires and G. Simon, "Dash in Twitch: Adaptive Bitrate Streaming in Live Game Streaming Platforms," in *Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming*, 2014, pp. 13–18.
[33] Y. Bejerano, C. Raman, C. Yu, V. Gupta, C. Gutterman, T. Young, H. Infante, Y. Abdelmalek, and G. Zussman, "DyMo: Dynamic monitoring of large scale LTE-Multicast systems," *IEEE INFOCOM*, May 2017.
[34] D. Maltz, J. Broch, D. Johnson, Y. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *ACM MobiCom*, vol. 114, 1998.
[35] J. Ikuno, M. Wrulich, and M. Rupp, "System level simulation of LTE networks," *IEEE 71st Vehicular Technology Conference*, May 2010.
[36] J. Quinlan, A. Zahran, and C. Sreenan, "Datasets for AVC (H.264) and HEVC (H.265) Evaluation of Dynamic Adaptive Streaming over HTTP (DASH)," *ACM Multimedia Systems Conference*, May 2016.
[37] N. Cranley, P. Perry, and L. Murphy, "User perception of adapting video quality," *International Journal of Human-Computer Studies*, vol. 64, no. 8, pp. 637–647, 2006.