

# Optimized Resource Distribution for Interactive TV Applications

Gosala Kulupana, *Student Member, IEEE*, Dumidu S. Talagala, *Member, IEEE*, Hemantha Kodikara Arachchi, *Member, IEEE*, and Anil Fernando, *Senior Member, IEEE*

**Abstract** — *This paper proposes a novel resource optimization scheme for cloud-based interactive television applications that are increasingly believed to be the future of television broadcasting and media consumption, in general. The varying distribution of groups of users and the need for on-the-fly media processing inherent to this type of application necessitates a mechanism to efficiently allocate the resources at both a content and network level. A heuristic solution is proposed in order to (a) generate end-to-end delay bound multicast trees for individual groups of users and (b) co-locate multiple multicast trees, such that a minimum group quality metric can be satisfied. The performance of the proposed heuristic solution is evaluated in terms of the serving probability (i.e., the resource utilization efficiency) and execution time of the resource allocation decision making process. It is shown that improvements in the serving probability of up to 50%, in comparison with existing resource allocation schemes, and several orders of magnitude reduction of the execution time, in comparison to the linear programming approach to solving the optimization problem, can be achieved<sup>1</sup>.*

**Index Terms** — **Interactive television, social multimedia applications, consumer video distribution, networking and computational resource optimization.**

## I. INTRODUCTION

The Television (TV), being one of the most widespread consumer devices of all-time, has contributed immensely to the unprecedented growth of the consumer entertainment sector. However, although display technologies have enabled a more realistic reproduction of content, the increasing use of mobile devices (e.g., mobile phones and tablets) and on-demand content consumption has resulted in the decline of the traditional broadcast medium. Innovative applications and services deployed on top of the existing TV infrastructure, including Interactive TV (ITV) [1], Social TV [2], personalized advertisements [3] and TV-based online gaming

[4] are envisaged as the next evolution of content delivery and consumption, where consumers actively engage with the content in a bi-directional TV concept using a variety of consumer devices (e.g., Kinect, Oculus Rift, Google Glass).

Anticipating the demands of these next generation ITV applications, enticing user interaction with the media by personalizing the audio-visual content for groups of like-minded viewers has been proposed [5]. Naturally, this results in more than one user actively engaging with the content; thus, the personalized television show will include multiple users, together, in the same scene. Furthermore, each group becomes part of a mini-TV show, and any number of such groups may exist and interact with the original TV show. An example of such a mini-show is one where two home users virtually appear in a sing-along musical program, watched by a group of peers. The user experience therefore becomes critically important for the successful adoption of this style of consumer application. On one hand, this implies that near real-time media processing and distribution is necessary, yet on the other, it is unviable to transfer the processing and routing intelligence to the consumer devices due to their limited awareness of the user group composition, network and computational capabilities. Thus, computational and network resource allocation process for ITV applications poses a unique set of challenges that cannot be realized through conventional resource allocation mechanisms.

In this context, the use of cloud infrastructure, as a means of delivering on-demand services to consumers [6], presents itself as an ideal mechanism to process and deliver the interactive content. In fact, cloud-based architectures have been proposed for personalizing broadcast media [7] and have been widely adopted within the ITV domain [4], [8], [9]. However, the increased physical separation between the service and the consumer will introduce additional latency [10] to the interactive application, in addition to the abstraction of physical resources in traditional clouds thereby limiting the flexibilities afforded to the application developer. In response, the cloud concept itself is undergoing dramatic changes at present (e.g., the emergence of inter-cloud and multi-cloud networking capabilities) [11]. Furthermore, these developments including the emerging Software Defined Networking (SDN) [12] concepts envisage a future where SaaS (Software as a Service) providers (e.g., streaming media applications) can dynamically move their application to a data

---

<sup>1</sup>This work was supported by the EU FP7 project ACTION-TV (<http://www.action-tv.net>), funded under the European Commission's 7th Framework Program (Grant Number: 611761).

The authors are with the Centre for Vision Speech and Signal Processing (CVSSP), Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, Surrey GU2 7XH, United Kingdom. Tel: +44 1483 686030, Fax: +44 1483 686031, E-mail: {g.kulupana, d.talagala, h.kodikaraarachchi, w.fernando}@surrey.ac.uk.

center of choice while specifying the routing paths of data flows. Thus, from the perspective of the consumer and interactive application developer, allocation of resources within this type of infrastructure must meet certain expectations; 1). The resource allocation process should be completed as soon as possible in order to cope with dynamically varying resources, 2). Each consumer should experience a superior quality irrespective of the location he/she resides, and 3). The latency between the hosted cloud and the consumer must be sufficiently small to facilitate the interactive nature of the application[13]. Considering these challenges, this paper proposes an efficient scheme to allocate computational and network resources for an ITV application, where the participants are in-home content consumers.

The remainder of this paper is organized as follows. First, a discussion of the state-of-the art in resource allocation for cloud-based systems is presented in Section II, followed by a formal definition of the problem and the optimization criteria in Section III. The proposed heuristic solution methodology and algorithms are described in Section IV, followed by the simulation configuration in Section V. Finally, the performance of the proposed methods are evaluated and compared with existing resource allocation methods in Section VI, and is followed by the concluding remarks in Section VII.

## II. RELATED WORK

In the context of the consumer ITV application described in the previous section, resource allocation must occur at both the application resource layer (e.g., processor and memory allocation) and the networking resource layer (e.g., network route selection). Although the allocation of cloud computing resources amongst competing tasks has been studied in the literature, these have predominantly focused on optimizing application layer resources alone. For example, Filali *et al.* have described a resource allocation scheme for grid computing infrastructure that maximizes a user's Quality of Service (QoS) [14], where QoS has been expressed as a function of the allocated resources in the application resource layer (i.e., the cloud). However, this implies that the effects of the network conditions and the actual content on the user's perception of the application are ignored. In order to include some of the factors that had been overlooked, specifically the content dependency, Nan *et al.* proposed an application layer resource allocation mechanism for multimedia applications [15], where a queuing based approach for different media tasks was adopted. The authors have also proposed several other resource allocation schemes [16], [17], [18], which are similar to [15] and adopt a First-In, First-Out (FIFO) approach to the queuing process. In each of these schemes, tasks in the queue are sequentially assigned the best processing resource using a greedy heuristic approach. Thus, although the queuing approach may well suit tasks of a short, bursty nature, a global optimum may not be attainable for continuous media tasks, such as in the personalized interactive video distribution scenario considered. The computational and networking resources allocation for a multi-group interactive application

has been proposed in one of our previous papers [5]. However all of these approaches assume that virtual links exist between data centers; hence, network layer resource optimization was implicitly disregarded.

However, the simultaneous optimization of application and network layer resource allocation is essential for the efficient content distribution in a continuous multimedia application. A number of works have previously attempted to address the simultaneous optimization problem. Gao *et al.* [19] modeled the objective function as a combination of emissions, energy costs and latency, and solved the problem using a combination of linear programming (LP) and a heuristic method. Although this method considers several parameters of interest, the computation of optimal routes through the network have not been considered. The cost-efficient resource allocation in cloud data centers considering the QoS requirements of the users, proposed by Hans *et al.* in [20], adopts a solution based on a Mixed Integer Linear Programming (MILP) solver and a LP relaxation technique, but suffers from the same drawback of neglecting the route selected through the network.

In the context of the consumer multimedia application considered in this paper, the simultaneous optimization of application and network layer resources (in a SDN for example) can therefore be distilled into a single problem; the optimal placement of processing nodes in a collection of cloud servers. In this regard, Larumbe *et al.* proposed a Tabu search based algorithm [21] to optimally allocate data centers to software components considering the routing optimization between the processing nodes, access nodes and backbone routers. This approach assigns each task to a processing node such that QoS can be maximized for specific processing requirements. In this scenario, the resource optimization problem is solved using either a MILP solver or the Tabu search heuristic approach using a greedy solution as an initial approximation of the optimal solution.

Although the resource optimization scheme proposed in [21] is the closest applicable to the ITV application scenario addressed in this paper, a number of significant drawbacks exist. Firstly, the optimization scheme in [21], as with most other methods in the literature, do not consider the multicasting nature of interactive consumer applications, i.e., when the same media is requested by several user domains. Secondly, the traditional QoS based solution does not sufficiently reflect the expectations and experience of geographically distributed user groups; thus, the formation of coherent groups of users, which is an important aspect of future interactive applications, is ignored. Thirdly, the computation time of the resource allocation process should support a real-time application. In order to satisfy the three requirements above, a centralized heuristic application and network layer resource allocation scheme is proposed in the following sections. Furthermore, it is assumed that the proposed scheme is supported by a SDN architecture, where a centralized controller senses and controls the virtual network [12] and determines the optimal route for each consumer based on his/her location, activity and available resources.

### III. PROBLEM FORMULATION

#### A. System Description

Fig. 1 illustrates an example logical network architecture diagram of an ITV distribution network. Here, two user groups are connected to two Internet Service Providers (ISPs) (i.e., access nodes). The four cloud computing resources (i.e., processing nodes) and three backbone routers (i.e., routing nodes) form the remainder of the network. Throughout the course of this work, it is assumed that this ITV distribution network satisfies the following requirements:

- (A.1). All nodes in the network support multicasting.
- (A.2). Users may join, withdraw or migrate from a particular user group and may create new groups of users.
- (A.3). A single processing node serves each user group.
- (A.4). Processing nodes may participate in the content distribution process.

Let  $G(V, E)$  represent the connected network where  $V = \{S, A, R\}$  is the set of nodes, including  $S = \{s_1, s_2, \dots, s_S\}$  the set of processing nodes (i.e., clouds),  $A = \{a_1, a_2, \dots, a_A\}$  the set of access nodes (i.e., ISPs) and  $R = \{r_1, r_2, \dots, r_R\}$  the set of routing nodes (i.e., routing devices) available in the network. Let  $E$  be the set of edges connecting different nodes and  $U = \{u_1, u_2, \dots, u_U\}$  be the set of ITV viewers uniquely belonging to the set of user groups  $N = \{n_1, n_2, \dots, n_N\}$ . The existence of a user  $u$  who belongs to the  $n^{\text{th}}$  ( $n \in N$ ) user group and connected to the  $a^{\text{th}}$  ( $a \in A$ ) access node is denoted by,

$$u \mapsto u_n^a = \begin{cases} 1 & \text{if } u \text{ is a member of social group } n \text{ and is} \\ & \text{connected to the access node } a, \\ 0 & \text{otherwise} \end{cases}$$

The main decision variables relevant to this resource allocation problem can be defined as follows:

$$x_{i,j}^{n,a} = \begin{cases} 1 & \text{if the edge from node } i \text{ to } j \text{ is used in the} \\ & \text{multicast tree for users in group } n \text{ who} \\ & \text{reside in access node } a, \\ 0 & \text{otherwise} \end{cases}$$

$$y_{i,j}^n = \begin{cases} 1 & \text{if the edge from node } i \text{ to } j \text{ is used in the} \\ & \text{multicast tree for social group } n, \\ 0 & \text{otherwise} \end{cases}$$

$$z^{n,s} = \begin{cases} 1 & \text{if processing node } s \text{ processes the media} \\ & \text{content of social group } n \\ 0 & \text{otherwise} \end{cases}$$

The required and available resources are denoted by:

- $B_n$  : Transmission bandwidth of the interactive media multicast required by the user group  $n$ .
- $B_{i,j}$  : Available bandwidth from node  $i$  to  $j$ .
- $b_e$  : Available bandwidth of edge  $e \in E$ .
- $P_n$  : Processing capability required by user group  $n$ .
- $p_s$  : Available processing power at processing node  $s$ .
- $D_{i,j}$  : Average link delay from node  $i$  to  $j$ .

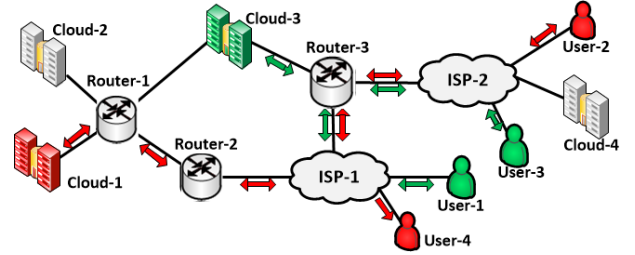


Fig. 1. A logical network architecture diagram of a consumer interactive TV distribution system. The diagram illustrates four users connected to two ISPs (access nodes) where “Cloud-1” and “Cloud-3” act as the media processing nodes of the two groups identified by the colours red and green, respectively. Media streams are multicasts traversing the network created by the collection of clouds, routers and ISPs forming a SDN.

- $\Delta$  : Maximum delay tolerable by the ITV application.
- $\Delta_{n,a}$  : Maximum delay tolerable from the processing node to the  $a^{\text{th}}$  access node for the  $n^{\text{th}}$  user group. (i.e.,  $\Delta_{n,a} = \Delta - \max(D_{a,u})$ , for all  $u$  in group  $n$  connected to  $a$ )

The various requirements imposed on the network (and, by extension, on the resource allocation problem) by the ITV application scenario can be listed as a set of constraints given below (these are further elaborated in the following paragraph). Thus, for each user group  $n$ , processing node  $s$ , routing node  $r$  and access node  $a$ ,

$$(C.1). \quad \sum_{j \in E_s^o} x_{s,j}^{n,a} - \sum_{j \in E_s^i} x_{i,s}^{n,a} = z^{n,s}$$

$$E_s^o \Rightarrow \{ \text{outgoing edge nodes from } s \} \text{ and } E_s^i \Rightarrow \{ \text{incoming edge nodes to } s \}$$

$$(C.2). \quad \sum_{j \in E_a^i} x_{i,a}^{n,a} = 1, E_a^i \Rightarrow \{ \text{incoming edge nodes to } a \}$$

$$(C.3). \quad \sum_{j \in E_r^o} x_{r,j}^{n,a} - \sum_{i \in E_r^i} x_{i,r}^{n,a} = 0$$

$$E_r^o \Rightarrow \{ \text{outgoing edge nodes from } r \} \text{ and } E_r^i \Rightarrow \{ \text{incoming edge nodes to } r \}$$

$$(C.4). \quad \sum_{n \in N} y_{i,j}^n \times B_n \leq B_{i,j}$$

$$(C.5). \quad \sum_{n \in N} z^{n,s} \times P_n \leq p_s$$

$$(C.6). \quad \sum_{(i,j) \in E} x_{i,j}^{n,a} \times D_{i,j} \leq \Delta_{n,a}$$

$$(C.7). \quad \sum_{s \in S} z^{n,s} = 1$$

$$(C.8). \quad \sum_{i \in E_j^i} y_{i,j}^n \leq 1, E_j^i \Rightarrow \{ \text{incoming edge nodes to } j \}$$

$$(C.9). \quad \forall a \in A \quad y_{i,j}^n \geq x_{i,j}^{n,a}$$

The constraint (C.1), above, describes the functionality of the processing nodes in the network. Here, if the node  $s$  serves the  $n^{\text{th}}$  user group, the media is streamed over a single outgoing edge in the set  $E_s^o$ , whereas if  $s$  functions as a relay node (as per (A.4)) the media also streams in over a single incoming edge in the set  $E_s^i$ . Similarly, (C.2) ensures that access nodes receive a single incoming media stream on its incoming edges denoted by  $E_a^i$ . It should be noted that the links denoted by  $x_{i,a}^{n,a}$  in (C.1) and (C.2) are logical links that are later mapped

to the physical link by constraint (C.9). The relaying function of the routing node  $r$  in (C.3) is similar in to that in (C.1). The constraint (C.4) ensures that the edge from  $i$  to  $j$  has sufficient bandwidth for the media streams of all user groups traversing that link, while (C.5) specifies that the processing node  $s$  has sufficient processing capacity to process all user groups allocated to it. In order to facilitate the interactive nature of the application, the transmission delay introduced by the network should not exceed a specified maximum of  $\Delta_{n,a}$ , as per (C.6).  $\Delta_{n,a}$  is obtained by subtracting the maximum delay of the  $n^{\text{th}}$  group's users connected to node  $a$  from the maximum tolerable delay  $\Delta$ . The constraint (C.7) ensures a single processing node  $s$  processes the  $n^{\text{th}}$  user group; thus, eliminating any synchronization issues that may arise when multiple users are engaging with the media content. Similarly, (C.8) ensures that the media stream of a particular user group is only received on one incoming edge; thus, satisfying (A.1). The relationship between  $x_{i,j}^{n,a}$ , the logical links, and  $y_{i,j}^n$ , the physical link, is defined by (C.9) and ensures that multiple logical transmissions carrying the same media stream is mapped to a single transmission in the physical media.

### B. Group QoS Cost Minimization

Each user's QoS can be modeled as the sum of end-to-end link QoS parameters from the processing node to the user. Here, a similar approach to Hyun *et al.* who proposed a QoS cost metric for IPTV systems [22] is adopted and extended to maintain an acceptable QoS during interactivity by the imposition of a delay bound. The link QoS cost metric for the ITV application can therefore be modelled (assuming media is transmitted at an approximately fixed rate) as

$$Qc_{i,j}^e = \alpha_1 \times (L_{i,j}) + \alpha_2 \times (J_{i,j}) + \alpha_3 \times (D_{i,j}) , \quad (1)$$

where  $Qc_{i,j}^e$  refers to the QoS cost of the edge from node  $i$  to  $j$ ,  $L_{i,j}$  refers to the average packet loss rate along the link from  $i$  to  $j$  and  $J_{i,j}$  refers to the jitter in the path.  $\{\alpha_1, \alpha_2, \alpha_3\}$  are constants parameterizing the QoS cost metric for an IPTV scenario [22]. The  $n^{\text{th}}$  user group's QoS cost from the processing node  $s$  to access node  $a$ , weighted by the number of users in the group, can now be expressed as

$$\varphi_{1,s}^{n,a} = \left( \sum_{(i,j) \in E} Qc_{i,j}^e \times x_{i,j}^{n,a} \right) \times \sum_{u \in U} u_n^a , \quad (2)$$

and the QoS cost from the access nodes to the users can be expressed as

$$\varphi_2^{n,a} = \left( \sum_{u \in U} Qc_u \times u_n^a \right) , \quad (3)$$

where  $Qc_u$  denotes the QoS cost from the user  $u \mapsto u_n^a$  to his access node. The  $n^{\text{th}}$  group's QoS cost can be defined as the summation of (2) and (3), normalized by the number of users in the group, as

$$\varphi_s^n = \frac{1}{\left( \sum_{a \in A} \sum_{u \in U} u_n^a \right)} \times \sum_{a \in A} \left\{ \varphi_{1,s}^{n,a} + \varphi_2^{n,a} \right\} . \quad (4)$$

Thus, the group QoS cost of the overall system becomes the summation of (4)  $\forall n$ . Therefore, minimizing the group QoS

cost implies,

$$\text{minimize} \left( \sum_{n \in N} \varphi_s^n \right) . \quad (5)$$

## IV. SOLUTION METHODOLOGY

Three main approaches exist to solve the optimization problem described in the previous section; linear programming methods, greedy resource allocation methods and heuristic methods. For completeness, first the two former approaches are briefly described below and the proposed heuristic solution approach is elaborated in the remainder of this section.

### A. Optimum Mixed Integer Linear Programming Method

The solution to the ITV distribution problem entails calculating an optimal resource allocation that satisfies (5) subject to the constraints (C.1) to (C.9). A solver which supports binary decision variables can be used to solve this problem since it satisfies the linear programming restrictions. In this paper, MATLAB toolboxes (YALMIP [23] and MOSEK) are used to model and compute a solution. The solver first executes a pre-solving operation which eliminates redundant constraints and variables. Next, several feasible solutions are approximated using a heuristic method prior to optimization using the 'branch-and-cut' method. The drawbacks of this style of MILP solvers are the large memory and processing time requirements. However, due to their theoretical ability to approach the optimal solution, their results are utilized as a benchmark for comparison purposes.

### B. Greedy Resource Allocation Methods

Greedy resource allocation is the simplest and most straightforward approach to allocate resources to competing user groups. In the context of the problem formulated in this paper, in a pure greedy approach, the multicast group would be created per user group and assigned to the network sequentially. However, once assigned these groups would be immovable (and therefore greedy) and reduce the available resources. Thus, later user groups would be more resource constrained, and result in a higher probability of failing to find the appropriate resources. Much of the related work in the literature adopt similar greedy approaches to different degrees [15]-[18], [24] and have therefore been used for the performance comparison of the proposed method. In addition, in the proposed method, a greedy allocation of resources can be considered during the multicast tree co-location phase (i.e., Step 3 of the heuristic method described below in Sec. IV.C). This approach is used as an additional comparison method, and is used to evaluate the effectiveness of the algorithms proposed in Step 2 of the heuristic method in Sec. IV.

### C. Proposed Heuristic Method

In order to overcome the shortcomings of the MILP and greedy resource allocation approaches a heuristic solution that satisfies the objective criteria is presented. The proposed solution mechanism consists of the following operations:

**Step 1:** Determine the delay bound, minimal cost path from each potential processing node to access node pair.

The method proposed by Salama *et al.* in [25] is applied, where the link cost  $Qc_{i,j}^e$  is considered for the scenarios described in Sec. III.B.

**Step 2:** Derive a set of multicast trees rooted at a particular processing node for each user group, subject to the objective criteria in (5). Apply Algorithm 1.

**Step 3:** Optimally co-locate each user group's multicast tree such that edge bandwidth and cloud processing limitations are satisfied. Apply the proposed multicast tree co-locating process in Algorithm 2.

#### 1) Multicast Tree Generation: Group QoS Cost Minimization

Algorithm 1 constructs the end-to-end multicast tree that minimizes the QoS cost (i.e., Step 2 of the proposed approach) of each user group. However, prior to its execution, for a user group  $n$  and processing node  $s$ , a delay bound minimum QoS cost unicast tree is constructed (from Step 1, as per [25]) to each access node  $a$  that connects the users of the  $n^{\text{th}}$  group.

The unicast trees computed in Step 1 act as inputs to Algorithm 1. During the initial resource allocation phase, for each user group  $n$ , multicast trees are constructed for every possible cloud  $s$  over a filtered network whose edge bandwidths exceed the transmission requirements of the user group. For each group, the existence of common intermediate nodes in the current unicast tree and partially completed multicast tree is evaluated. If no such nodes exist, the current unicast tree is integrated to the multicast tree without any alteration. However, if such nodes do exist, the function MULTICAST\_REDUCTION\_QOS is executed to amend the unicast tree to be compliant with the multicast tree. Furthermore the proposed approach can adapt to varying network conditions; thus, the need for complete resource re-allocation is eliminated. Here the resource allocation is performed only for user groups which are affected by the network change (i.e., for those whose bandwidth and the delay requirements are not satisfied), whereas remaining groups remain unaltered. The dynamic allocation is identified by setting the flag “*dynamic\_flag*”, which, when set gives preference to the “*initial\_cloud*” (i.e., the processing cloud of the current user group before the network parameters change) in the multicast tree generation process.

In the procedure MULTICAST\_REDUCTION\_QOS, for a common intermediate node  $r$  (starting from the closest to the processing node  $s$ ), the function evaluates the various unicast paths from  $s$  to  $r$ . If these are identical, they form part of the multicast tree from  $s$  to  $r$ . In the event that they are not, an alternate path which satisfies the delay requirement of all the users with the minimum overhead is selected. This is motivated by the following scenario; e.g., if an access node  $a_1$  is located closer to  $s$ , the minimum end-to-end QoS cost could be achieved using a longer, higher delay path. However this is not possible for an access node  $a_2$  further away from  $s$ , since the increasing number of hops introduce a natural delay constraint, thereby violating the delay requirement of the ITV application for users in  $a_2$ . Thus, the selection of the minimum delay path for the multicast tree enables serving both access nodes  $a_1$  and  $a_2$ , albeit at an increased QoS cost to  $a_1$ . The

creation of the multicast path is done sequentially (for each node  $a$ ), while considering the cost of changing the multicast path. Here, the cost of altering the multicast tree is compared with that of using an alternate route to a particular access node. If the cost of re-routing the multicast tree is greater, the alternate route for the access node is adopted. This results in a time complexity in the order of  $O(|S| |A| |V|^3)$  for Algorithm 1.

#### 2) Multicast Tree Co-Location

Algorithm 2 proposes a heuristic method to co-locate multicast trees (i.e., Step 3 of the proposed approach) generated in Step 2. The proposed approach is influenced by the group multicasting concepts proposed in [26] and [27], and addresses the simultaneous consideration of both networking and processing constraints when co-locating multicast trees.

The operation of the Algorithm 2 is as follows. First, the multicast trees obtained from Algorithm 1 are co-located sequentially, in order of decreasing required bandwidth. In the event that neither processing nor networking constraints are violated, this represents the minimum cost group multicast tree. However, in the event that saturated links or nodes are encountered, the multicast trees are re-routed by calling the function REROUTE\_MULTICAST\_TREES.

This function evaluates the cost of two possible outcomes; re-routing the last user group sequentially added to the group multicast tree, or re-routing all user groups that utilize the saturated resources. In each case, the re-routing costs are represented by the marginal costs of newly computed alternate multicast trees, obtained by executing Step 2 (i.e., compute alternate delay bound minimum cost multicast trees) once more, excluding the links and nodes that became saturated. In the event that the re-routing cost of the latest user group exceeds the re-routing cost of each existing user group, the existing user group with least overhead is re-routed, and the latest user group is re-routed otherwise. Thus, the allocation of the resources to the user groups is no longer completely greedy. This results in a more optimal use of resources as demonstrated by the simulation results in the following sections. It should also be noted that the heuristic group multicast tree co-location method proposed here, is therefore immediately applicable to a dynamic scenario where new user groups join or exit the network at different times. The time complexity of Algorithm 2 is in the order of  $O(|N| |E| |M|)$ , where  $M$  represents the complexity of Step 2 (Algorithm 1).

## V. SIMULATION ENVIRONMENT

The performance of the proposed heuristic resource allocation technique is evaluated in MATLAB, using 200 Monte Carlo simulations of different network conditions. The simulations are carried out on a 32 core Dell PowerEdge R710 (8 Intel Xeon Quad-Core E5520 2.2GHz processors) server with 144 GB memory. The resource requirements are prescribed by the interactive, personalized video distribution application described in Sec. I. For simplicity, HD HEVC transmission is assumed for each user group with a bandwidth of 8 Mb/s [22]. In order to maintain an acceptable perceived quality, the maximum allowable interaction delay is restricted

---

**Algorithm 1.** Multicast tree generation for group QoS cost minimization.

---

$uni\_tree(n,s,a) \leftarrow$  Step 1: Compute the delay bound minimal QoS cost unicast trees from  $\forall s$  for  $n$ .

**procedure** MULTICAST\_TREE\_QoS ( $G,U,\{Qc_{i,j}^e\},uni\_tree,$   
 $n,initial\_cloud,dynamic\_flag$ )

**for**  $s \in S$  (where  $p_s > P_n$ )

**if** (dynamic\_flag= true **and**  $s \neq initial\_cloud$ )

$\varphi_{1,s}^{n,a} \leftarrow c_{max}$  % Force current cloud to remain unchanged

**end if**

**for**  $a \in A$

**if** ( $a$  has users belonging to user group  $n$ )

$i\_nodes \leftarrow$  Find common intermediate nodes in  $uni\_tree(n,s,a)$  along the path from  $s$  to  $a$ .

**if** ( $i\_nodes$  do not exist)

$mult\_tree(n,s) \leftarrow$  Path from  $s$  to  $a$  forms part of the multicast tree.

**else**

$mult\_tree(n,s) \leftarrow$  MULTICAST\_REDUCTION\_QoS

**end if**

Calculate QoS cost contributions from  $s$  to  $a$  along the path in  $mult\_tree$ , and from  $a$  to the end users.

$\varphi_{1,s}^{n,a} \leftarrow \varphi_{1,s}^{n,a} + Qc_{s,a}^{mult\_tree} \times \left( \sum_{u \in U} u_n^a \right)$

$\varphi_2^{n,a} \leftarrow \left( \sum_{u \in U} Qc_u \times u_n^a \right)$

**end if**

**end for**

Compute the QoS cost at node  $s$  for user group  $n$ .

$\varphi_s^n = \frac{1}{\left( \sum_{a \in A} \sum_{u \in U} u_n^a \right) \times \sum_{a \in A} \left\{ \varphi_{1,s}^{n,a} + \varphi_2^{n,a} \right\}}$

**end for**

$\varphi_n \leftarrow \min(\varphi_s^n)$

$mult\_tree \leftarrow mult\_tree(n,s_0)$ ;  $s \equiv s_0$  corresponds to the minimum QoS cost processing node.

**return**  $mult\_tree, \varphi_n$

**end procedure**

**procedure** MULTICAST\_REDUCTION\_QoS( $uni\_tree, i\_nodes,$   
 $mult\_tree, a, n$ )

**for**  $r \in i\_nodes$

**if** ( $s$  to  $r$  path in  $uni\_tree(n,s,a)$  and  $mult\_tree$  are common)

$mult\_tree(n,s) \leftarrow$  Assign as multicast path from  $s$  to  $r$

**else**

$path\_delays \leftarrow$  Calculate delay along path from  $s$  to  $r$  in  $uni\_tree(n,s,a)$  and  $mult\_tree$ .

**if** ( $path\_delay(mult\_tree) < path\_delay(uni\_tree)$ )

No change in multicast path from  $s$  to  $r$ .

$mult\_tree \leftarrow$  Add path from  $r$  to  $a$  from  $uni\_tree$ .

**else**

Compute the cost of change in the multicast path.

$a_m \leftarrow$  Cost of changing  $mult\_tree$  path from  $s$  to  $r$  to  $uni\_tree$ .

$a_a \leftarrow$  Cost of alternative path (excluding  $r$ ) from  $s$  to  $a$ .

**if** ( $a_m > a_a$ )

$mult\_tree \leftarrow$  Assign alternative  $s$  to  $r$  path for  $a$

**else**

Set  $s$  to  $r$   $mult\_tree$  path to  $uni\_tree$  path.

$mult\_tree \leftarrow uni\_tree$  path from  $s$  to  $r$ .

**end if**

**end if**

**end if**

**end for**

**return**  $mult\_tree$

**end procedure**

---

to ( $\Delta = 100$  ms) [13], and  $\Delta_{n,a}$  is derived accordingly. The network is assumed to be made up of 10 ISPs (access nodes), 10 cloud computing resources (processing nodes) and 10 routing nodes (backbone routers). In order to evaluate the effectiveness of the proposed algorithms, the interconnections and network loading of these resources are varied as follows. The number of interconnections between nodes of the network is determined at random, however, the available bandwidth, link latency, jitter and packet loss is restricted to be within 20 Mb/s  $< B_{i,j} < 60$  Mbps, 20 ms  $< D_{i,j} < 60$  ms, 5 ms  $< J_{i,j} < 60$  ms, 0.01%  $< L_{i,j} < 0.1\%$ , respectively [28]. The link latency between each user and his ISP (access node) is a uniform random variable in the interval (10 ms, 20 ms).

## VI. RESULTS AND DISCUSSION

The performance of the proposed heuristic algorithm is discussed in the following section. Two variants of the proposed algorithm are evaluated; the complete proposed algorithm (Steps 1 to 3 in Sec. IV.C) denoted by ‘‘Proposed (Heuristic)’’, and greedy multicast tree co-location in the proposed method (Steps 1 to 2 in Sec. IV.C and greedy

multicast tree co-location instead of Step 3) denoted by ‘‘Proposed (Partially Greedy)’’. The performance of these methods is compared with the MILP approach to solving the optimization problem (Sec. IV.A) and other existing resource allocation schemes. The MILP approach is limited to the first eight user groups, due to the increasing memory and execution time required for a larger number of user groups.

The performance of the proposed technique, where group QoS cost is minimized, is summarized and compared in Table I. In order to facilitate a fair comparison, the average group QoS cost and execution time is computed from the Monte Carlo simulations, where all presented approaches achieve a feasible solution. As expected, the results reveal an increasing group QoS cost and execution time with the number of user groups. In addition, the minimum group QoS cost is achieved by the MILP approach, whereas the maximum corresponds to the greedy approach. The performance of the proposed method is confined between these extremes, with the ‘‘Proposed (Partially Greedy)’’ approach exhibiting a higher cost. The improved performance of the ‘‘Proposed (Heuristic)’’ method can be mainly attributed to the cost based re-routing of the multicast trees during the co-location process in Algorithm

---

**Algorithm 2.** Multicast tree co-location.

---

```

procedure MULTICAST_TREE_CO-LOCATION ( mult_tree )
  combined_trees={ }
  for  $n \in N$ 
    temp_combined_trees  $\leftarrow$  Include mult_tree( $n$ )
    for  $e \in E$ 
      if ( edge  $e$  bandwidth in temp_combined_trees  $>$   $b_e$  )
         $E^i \leftarrow$  Note  $e$  as saturated, add to saturated edge set.
         $N^e \leftarrow$  Store user groups using the saturated link  $e$ ,
          excluding user group  $n$ .
      end if
    end for
  for  $s \in S$ 
    if ( $p_s <$  processing requirements of the combined set of user
      groups in temp_combined_trees )
       $S^i \leftarrow$  Note processing node  $s$  as saturated.
       $N^s \leftarrow$  Store user groups which use the saturated node  $s$ ,
        excluding user group  $n$ .
    end if
  end for
  if ( $E^i$  or  $S^i$  exist)
    combined_trees  $\leftarrow$  REROUTE_MULTICAST_TREES
  else
    combined_trees  $\leftarrow$  temp_combined_trees.
  end if
end for
return combined_trees
end procedure

```

```

procedure REROUTE_MULTICAST_TREES( $E^i$ ,  $N^e$ ,  $S^i$ ,  $N^s$ , mult_tree,
  temp_combined_trees,  $n$ )
// Calculate re-routing cost of existing combined_trees
for  $e^i \in E^i$ 
  for  $n^e \in N^e$  not re-routed by a previous  $e^i$ 
    1. Refine network by eliminating resources used by user
      groups in temp_combined_trees except  $n^e$ .
    2. Eliminate links from  $G$  that do not satisfy the minimum
      bandwidth required by user group  $n^e$ .
    3. Compute alternative multicast tree for  $n^e$  using the
      refined network (re-run Step 2).
    4.  $C_e(n^e) \leftarrow$  Cost of re-routing (i.e., cost difference
      between original and alternative multicast trees).
  end for
end for

```

```

 $n^p(e^i) \leftarrow$  Select  $n^e$  corresponding to the  $\min(C_e)$ , and retain as a
  potential user group for re-routing.
temp_combined_trees  $\leftarrow$  Alternate multicast tree of  $n^p$ 
 $E^i \leftarrow$  Remove edges no longer saturated from  $E^i$ .
 $S^i \leftarrow$  Remove nodes no longer saturated from  $S^i$ 
 $N^s \leftarrow$  Remove  $n^p$  from  $N^s$  if  $n^p$  exists in  $N^s$ .
end for
if ( $S^i \neq \phi$ )
  for  $n^s \in N^s$ 
    1. Refine network by eliminating resources used by user
      groups in temp_combined_trees except  $n^s$ .
    2. Eliminate links from  $G$  that do not satisfy the minimum
      bandwidth required by user group  $n^s$ .
    3. Compute alternative multicast tree for  $n^s$  using the
      refined network (re-run Step 2).
    4.  $C_s(n^s) \leftarrow$  Calculate cost of re-routing (i.e., cost
      difference between original and alternative multicast
      trees). Assign infinite cost if  $P_{n^s} < P_n$ .
  end for
   $n^p(S^i) \leftarrow$  Select  $n^s$  corresponding to the  $\min(C_s)$ , and retain as a
    potential user group for re-routing.
  temp_combined_tree  $\leftarrow$  Alternate multi. tree of  $n^p(S^i)$ 
end if
// Calculate re-routing cost of user group  $n$ .
  1. Refine network by eliminating resources used by the user
    groups in combined_trees.
  2. Eliminate links from  $G$  that do not satisfy the minimum
    bandwidth required by user group  $n$ .
  3. Compute alternative multicast tree for  $n$  using the refined
    network (re-run Step 2).
  4.  $C_n \leftarrow$  Cost of re-routing (i.e., cost difference between original
    and alternative multicast trees).
if (  $\sum_{e^i \in E^i} C_e(n^p(e^i)) + C_s(n^p(S^i)) > C_n$  )
  combined_trees  $\leftarrow$  Assign alternate multi. tree for  $n$ 
else
  Re-route saturated nodes already in combined_trees
  combined_trees  $\leftarrow$  temp_combined_trees
end if
return combined_trees
end procedure

```

**TABLE I**  
COMPARISON OF AVERAGE GROUP QoS COST AND AVERAGE EXECUTION TIME FOR THE GROUP QoS COST MINIMIZATION SCENARIO.

| User Groups        |            | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   | 11   | 12   | 13   | 14   | 15   |
|--------------------|------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Group QoS Cost     | MILP       | 0.22 | 0.40 | 0.66 | 0.90 | 1.12 | 1.34 | 1.51 | 1.63 | -    | -    | -    | -    | -    | -    | -    |
|                    | Prop. (H)  | 0.22 | 0.44 | 0.68 | 0.90 | 1.13 | 1.37 | 1.58 | 1.85 | 2.08 | 2.29 | 2.54 | 2.80 | 3.05 | 3.30 | 3.55 |
|                    | Prop. (PG) | 0.22 | 0.44 | 0.68 | 0.91 | 1.14 | 1.38 | 1.61 | 1.84 | 2.10 | 2.30 | 2.56 | 2.78 | 3.08 | 3.32 | 3.57 |
|                    | Greedy     | 0.24 | 0.46 | 0.72 | 0.94 | 1.18 | 1.41 | 1.67 | 1.91 | 2.23 | 2.33 | 2.57 | 2.83 | 3.02 | 3.66 | 3.85 |
| Execution Time (s) | MILP       | 182  | 340  | 503  | 672  | 849  | 1009 | 1157 | 1367 | -    | -    | -    | -    | -    | -    | -    |
|                    | Prop. (H)  | 1    | 1    | 1    | 3    | 4    | 5    | 6    | 8    | 8    | 13   | 13   | 15   | 12   | 18   | 27   |
|                    | Prop. (PG) | 1    | 1    | 1    | 2    | 2    | 2    | 2    | 2    | 2    | 3    | 3    | 4    | 5    | 5    | 5    |
|                    | Greedy     | 1    | 1    | 1    | 1    | 2    | 2    | 2    | 2    | 2    | 2    | 3    | 3    | 3    | 4    | 4    |

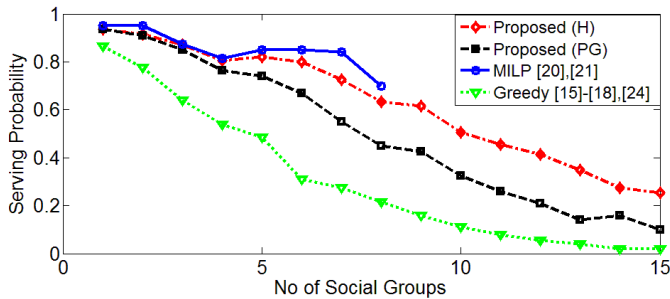


Fig. 2. Serving probability of all user groups in the system for the group QoS cost minimization scenario.

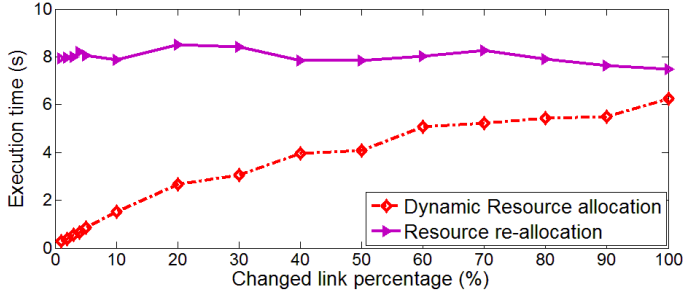


Fig. 3. Execution time of the proposed resource allocation scheme in a dynamic networking scenario for a fixed user configuration.

2. Although the disparity in the group QoS cost (in relative terms) is minimal between the optimization approaches, significant differences in execution times are observed. The MILP approach exhibits several orders of magnitude greater execution times compared to the three other methods, which can be attributed to the non-convex nature of the optimization problem and the related complexity of the solver. The proposed methods' execution times are comparable to each other, as well as that of the greedy resource allocation method.

Fig. 2 illustrates the serving probability (i.e., the likelihood of a particular approach finding a feasible solution) of the four approaches. As expected, the serving probability decreases with increasing numbers of user groups, and the MILP approach achieves the highest serving probability due to near exhaustive search method adopted. The "Proposed (Partially Greedy)" method achieves better performance over the greedy approach due to the re-arranging during multicast tree creation in Algorithm 1. This is further enhanced by the application of Algorithm 2 in the "Proposed (Heuristic)" method, resulting in an up to 50% increase in the serving probability. It should be noted that although this is primarily due to the re-routing of multicast trees during the co-location process, yet the impact on the serving probability in Fig. 2 is much more pronounced in comparison to its effect on the group QoS cost in Table I.

Fig.3 and Table II illustrate the capacity of the proposed method to adapt to varying network conditions. Both methods illustrated utilize the proposed multicasting approach, however in the re-allocation method, a complete resource allocation occurs, whereas in the dynamic allocation method preference is given to the initial processing nodes. The results are obtained for the same 200 network and user configurations used previously, where the network parameters of a random set of edges are allowed to change dynamically. The results indicate that complete re-allocation affects a greater percentage of users (a user is assumed to be affected when his/her serving cloud changes over time) and consumes a substantial amount of time on computation. In contrast, the proposed dynamic resource allocation in Algorithm 2 can achieve better performance (reduced execution time and disruption to users), albeit at a marginal increase of the group QoS cost as shown in Table II.

## VII. CONCLUSION

In this paper, an efficient scheme to allocate computational and network resources in a next generation ITV application is proposed. First, the assumptions and constraints applicable to this application are described, and the cost function which maximizes end user QoS is derived. Next, a heuristic solution is proposed to this resource allocation problem (in terms of this cost function) that consists of two algorithms for end-to-end QoS cost minimized multicast tree generation and dynamic multiple multicast tree co-location. Multiple Monte Carlo trials of different network and user configurations were simulated to evaluate the proposed method's performance, and were compared with results obtained from the optimal MILP and existing greedy resource allocation approaches. The simulation results suggest that the proposed method can achieve comparable performance to the MILP approach, with several orders of magnitude reduction in the computational time required. In addition, an improvement in the ability to find a feasible resource allocation configuration of up to 50% is observed with respect to the greedy approaches.

## REFERENCES

- [1] F. M. Matsubara and M. Kawamori, "Lightweight interactive multimedia environment for TV," *IEEE Trans. Consum. Electron.*, vol. 57, no. 1, pp. 283–287, Feb. 2011.
- [2] E. Mantzari, G. Lekakos, and A. Vrechopoulos, "Social tv: introducing virtual socialization in the tv experience," in *Proc. of the 1st Int. conf. on Designing Interactive User Experiences for TV and Video - uxtv '08, Silicon Valley, California*, 2008, pp. 81–84.
- [3] T. Pessemier, T. Deryckere, K. Vanhecke, and L. Martens, "Proposed architecture and algorithm for personalized advertising on iDTV and

TABLE II

AVERAGE GROUP QoS COST AND UNAFFECTED USER PERCENTAGE COMPARISON FOR DYNAMIC AND COMPLETE RE-ALLOCATION OF RESOURCES.

| Changed link percentage (%) |                    | 1    | 2    | 3    | 4    | 5    | 10   | 20   | 30   | 40   | 50   | 60  | 70   | 80   | 90   | 100  |
|-----------------------------|--------------------|------|------|------|------|------|------|------|------|------|------|-----|------|------|------|------|
| QoS Cost                    | Dynamic allocation | 1.85 | 1.85 | 1.85 | 1.85 | 1.86 | 1.88 | 1.91 | 1.95 | 1.93 | 1.95 | 1.9 | 1.94 | 1.95 | 1.96 | 1.94 |
|                             | Re-allocation      | 1.84 | 1.84 | 1.83 | 1.84 | 1.85 | 1.83 | 1.83 | 1.86 | 1.81 | 1.8  | 1.8 | 1.8  | 1.78 | 1.81 | 1.81 |
| Unaffected users (%)        | Dynamic allocation | 99   | 98.3 | 97.4 | 97.6 | 96.4 | 92   | 85.9 | 81   | 75.2 | 73.6 | 66  | 64.2 | 64.1 | 61.9 | 57.9 |
|                             | Re-allocation      | 94.2 | 86   | 81   | 79.9 | 73.9 | 65.4 | 50.8 | 47.4 | 38.3 | 35   | 34  | 30.6 | 28.4 | 27.6 | 25.9 |



- mobile devices,” *IEEE Trans. Consum. Electron.*, vol. 54, no. 2, pp. 709–713, May 2008.
- [4] N. Veljkovic, M. Punt, M. Z. Bjelica, and N. Crvenkovic, “TV-centric multiplayer gaming over the cloud for consumer electronic devices,” in *2013 IEEE Third International Conference on Consumer Electronics Berlin (ICCE-Berlin)*, 2013, pp. 1–3.
- [5] G. Kulupana, D. S. Talagala, H. K. Arachchi, and A. Fernando, “Optimized resource distribution for interactive TV applications,” in *2015 IEEE International Conference on Consumer Electronics (ICCE)*, 2015, pp. 70–71.
- [6] W. Zhu, C. Luo, J. Wang, and S. Li, “Multimedia cloud computing,” *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 59–69, 2011.
- [7] SeungGwan Lee, Daeho Lee, and Sungwon Lee, “Personalized DTV program recommendation system under a cloud computing environment,” *IEEE Trans. Consum. Electron.*, vol. 56, no. 2, pp. 1034–1042, May 2010.
- [8] G. M. Calixto, A. C. B. Angeluci, L. C. P. Costa, R. de Deus Lopes, and M. K. Zuffo, “Cloud computing applied to the development of global hybrid services and applications for interactive TV,” in *2013 IEEE International Symposium on Consumer Electronics (ISCE)*, 2013, pp. 283–284.
- [9] M. Doke, H. Kaneko, N. Hamaguchi, and S. Inoue, “Engaging Viewers Through the Connected Studio: Virtual Participation in TV Programs,” *IEEE Consum. Electron. Mag.*, vol. 1, no. 4, pp. 30–39, Oct. 2012.
- [10] T. Hofbeld, R. Schatz, M. Varela, and C. Timmerer, “Challenges of QoE management for cloud applications,” *IEEE Commun. Mag.*, vol. 50, no. 4, pp. 28–36, 2012.
- [11] K. Hwang, G. Fox, and J. J. Dongarra, “Inter-cloud Resource Management,” in *Distributed and Cloud Computing*, San Francisco: Morgan Kaufmann Publishers, 2011, p. 246.
- [12] H. Owens and A. Durrresi, “Explicit Routing in Software-Defined Networking (ERSDN): Addressing Controller Scalability,” in *2014 17th International Conference on Network-Based Information Systems*, 2014, pp. 128–134.
- [13] J. Nielsen, *Usability Engineering*, vol. 44, no. 3. Morgan Kaufmann, 1993.
- [14] A. Filali, A. S. Hafid, and M. Gendreau, “Adaptive resources provisioning for grid applications and services,” in *IEEE Int. Conf. on Commun., Beijing, China*, 2008, pp. 186–191.
- [15] X. Nan, Y. He, and L. Guan, “Optimal Resource Allocation for Multimedia Application Providers in Multi-site Cloud,” in *IEEE Int. Symp. on Circuits and Syst., Beijing, China*, 2013, pp. 449–452.
- [16] X. Nan, Y. He, and L. Guan, “Towards Dynamic Resource Optimization For Cloud-Based Free Viewpoint Video Service,” in *Proc. of IEEE Int. Conf. on Image Process., Paris, France*, 2014.
- [17] X. Nan, Y. He, and L. Guan, “Optimal allocation of virtual machines for cloud-based multimedia applications,” in *IEEE 14th Int. Workshop on Multimedia Signal Process., Banff, Canada*, 2012.
- [18] X. Nan, Y. He, and L. Guan, “Optimal resource allocation for multimedia cloud based on queuing model,” in *IEEE Int. workshop on Multimedia Signal Process., Hangzhou, China*, 2011, pp. 1–6.
- [19] P. Gao, A. Curtis, B. Wong, and S. Keshav, “It’s not easy being green,” in *Proc. of the ACM SIGCOMM 2012 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Commun., Helsinki, Finland*, 2012, pp. 211–222.
- [20] R. Hans, U. Lampe, M. Pauly, and R. Steinmetz, “Cost-Efficient Capacitation of Cloud Data Centers for QoS-Aware Multimedia Service Provision,” in *4th Int. Conf. on Cloud Computing and Services Sci., Barcelona, Spain*, 2014, pp. 158–163.
- [21] F. Larumbe and B. Sanso, “A Tabu Search Algorithm for the Location of Data Centers and Software Components in Green Cloud Computing Networks,” *IEEE Trans. Cloud Comput.*, vol. 1, no. 1, pp. 22–35, 2013.
- [22] H. Kim and S. Choi, “A study on a QoS/QoE correlation model for QoE evaluation on IPTV service,” in *12th Int. Conf. on Advanced Commun. Technology, Phoenix Park, Ireland*, 2010, pp. 1377–1382.
- [23] J. Lofberg, “YALMIP: a toolbox for modeling and optimization in MATLAB,” in *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, 2004, pp. 284–289.
- [24] K. Sembiring and A. Beyer, “Dynamic resource allocation for cloud-based media processing,” in *Proc. of the 23rd ACM Workshop on Network and Operating Syst. Support for Audio and Video, New York*, 2013, pp. 49–54.
- [25] H. F. Salama, D. S. Reeves, and Y. Viniotis, “A distributed algorithm for delay-constrained unicast routing,” in *IEEE INFOCOM ’97, Kobe, Japan*, 1997, pp. 84–91.
- [26] Y. L. Wang, “Based QoS constrained group multicast routing for multimedia communication,” in *2010 Int. Conf. on Computer, Mechatronics, Control and Electronic Engineering, CMCE 2010, Changchun, China*, 2010, pp. 296–299.
- [27] C. P. Low and X. Song, “On finding feasible solutions for the delay constrained group multicast routing problem,” *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 581–588, 2002.
- [28] S. Ravot, Y. Xia, D. Nae, X. Su, and H. Newman, “A practical approach to TCP high speed WAN data transfers,” in *Proc. of PATHNets 2004 (First Workshop on Provisioning and Transport for Hybrid Networks, San José)*, 2004.

## BIOGRAPHIES



**Gosala Kulupana** (M’14-S’15) received his B.Sc. (Eng.) degree with honours from University of Moratuwa, Sri Lanka in 2011. From 2011 to 2014, he was an Engineer at Mobitel (pvt) Ltd., Sri Lanka. He is currently pursuing his Ph.D. in the Centre for Vision, Speech and Signal Processing (CVSSP) at the University of Surrey, United Kingdom.

His research interests are in the areas of resource optimization, distributed processing and error

resilience video coding.



**Dumidu S. Talagala** (S’11-M’14) received the B.Sc. Eng (Hons) in Electronic and Telecommunication Engineering from the University of Moratuwa, Sri Lanka, in 2007. From 2007 to 2009, he was an Engineer at Dialog Axiata PLC, Sri Lanka. He completed his Ph.D. degree within the Applied Signal Processing Group, College of Engineering and Computer Science, at the Australian National University, Canberra, in 2013.

He is currently a research fellow in the Centre for Vision, Speech and Signal Processing at the University of Surrey, United Kingdom. His research interests are in the areas of sound source localization, spatial soundfield reproduction and convex optimization techniques.



**Hemantha Kodikara Arachchi** (M’02) received his B.Sc. (Eng.) degree with honours and M.Phil. degree in Electronic and Telecommunication Engineering from University of Moratuwa, Sri Lanka in 1997 and 2000 and the Ph.D. degree in Telecommunications from AIT, 2004.

At present, he is a Senior Research Fellow at the CVSSP Multimedia Communications Research group of the University of Surrey, UK. His research interests are in video coding, video communication, QoE and context-aware content adaptation. He has published over 70 peer reviewed journal and conference papers.



**Anil Fernando** (S’98-M’01-SM’03) received the B.Sc. Engineering degree (First class) in Electronic and Telecommunications Engineering from the University of Moratuwa, Sri Lanka in 1995 and the MEng degree (Distinction) in Telecommunications from Asian Institute of Technology (AIT), Bangkok, Thailand in 1997. He completed his PhD at the Department of Electrical and Electronic Engineering, University of Bristol, UK in February 2001.

Currently, he is a reader in signal processing at the University of Surrey, UK. Prior to that, he was a senior lecturer in Brunel University, UK and an assistant professor in AIT. His current research interests include cloud communication, Quality of Experience, intelligent video encoding for wireless systems and video communication in LTE. He has published more than 290 international publications on these areas. He is a senior member of IEEE and a fellow of the HEA, UK.