

Optimal Incentive Design for Cloud-enabled Multimedia Crowdsourcing

Sabita Maharjan, *Member, IEEE*, Yan Zhang, *Senior Member, IEEE*, and Stein Gjessing, *Member, IEEE*

Abstract—Multimedia crowdsourcing possesses a huge potential to actualize many new applications that are expected to yield tremendous benefits in diverse fields including environment monitoring, emergency rescues during natural catastrophes, online education, sports and entertainment. Nonetheless, multimedia crowdsourcing unfolds new challenges such as big data acquisition and processing, more stringent QoS requirements, and heterogeneity of crowdsensors. Consequently, incentive mechanisms specifically tailored to multimedia crowdsourcing applications need to be developed to fully utilize the potential of multimedia crowdsourcing. In this paper, we design an optimal incentive mechanism for the smartphone contributors to participate in a cloud-enabled multimedia crowdsourcing scheme. We establish a condition that determines whether the smartphones are eligible to participate, and provide a close form expression for the optimal duration of service from the contributors, for a given reward from the crowdsourcer. Consequently, we derive the conditions for existence of an optimal reward for the contributors from the crowdsourcer, and prove its uniqueness. We numerically illustrate the performance of our model considering logarithmic and linear cost functions for the cloud resources. The similarity of the results for different cost models corroborate the validity of our model and the results, whereas the difference in the magnitudes suggest that the strategy of the crowdsourcer as well as the strategies of the smartphone participants considerably depend on the cloud cost model.

I. INTRODUCTION

Crowdsourcing [1], a practice of obtaining the needed data or other services by soliciting contributions from the crowd, has emerged as a method for distributed problem solving through participatory contributions from individuals. The potential of crowdsourcing has been extensively studied, e.g., for intelligent public transportation [2], vehicular communication [3], sensing the community health state [4], shop profiling [5], big data and homeland security [6] and recommender systems for social networking [7].

With the rapid proliferation of advanced mobile devices such as smartphones and tablets, they have become a pervasive platform for participatory sensing. Today, an average smartphone is typically equipped with a rich set of sensors, such as a microphone, a camera, an accelerometer and a GPS. As smartphones have seen advanced technology, design and functionalities, they can reveal the full potential of crowdsourcing through users' contribution to complex and novel problem solving for diverse applications including multimedia. Several

existing works have investigated applications focussing specifically on smartphones based crowdsensing, e.g., [8], [9], [10], [11].

A crowdsourcer is the entity that needs data or service(s), and that recruits the participants to provide it the data or the service(s). Individuals that sense, collect, and/or analyze, and share the data, or provide the service, are called the crowdsensors. Crowdsensing is the process whereby individuals and communities make use of their devices such as smart phones and tablets for getting the required data or the tasks completed. Participatory sensing is closely related to mobile crowdsensing. In participatory sensing, individuals are actively involved in sharing their sensor data. Mobile crowdsensing is an evolution of participatory sensing but mobile crowdsensing may involve both implicit and explicit participation, thereby mobile crowdsourcing can involve collecting data from both mobile sensing and mobile social networks [12], [13], [14]. For the smartphone crowd that generates and sends the multimedia data to the crowdsourcer, various terms such as crowdsensors and contributors are commonly used. In this paper, we use the term *contributor*, to represent the sources that generate multimedia for the crowdsourcer.

Multimedia such as image, audio and video, is an increasingly significant source of sensory observations and information about activities, incidents and the environment. The immense growth in the popularity of video streaming applications such as Youtube [15] over the Internet [16] indicates the possible increased use of multimedia traffic in future networks. Multimedia crowdsourcing possesses a huge potential to actualize many new applications that are expected to yield tremendous benefits in diverse fields including environment monitoring, online education, social networking, sports and entertainment and emergency rescues during and after natural disasters. For instance, multimedia crowdsourcing for forest monitoring can reveal details that usual sensor measurements can not, e.g., in the case of fire in the forest, the number of people trapped in a particular side or area, and other details. RapidShare, a file hosting system where anyone could upload and distribute files including those with multimedia, was once amongst the top 20 most visited websites [17]. In addition to entertainment, YouTube is a great platform for teaching diverse skills and techniques online. Similarly, facebook [18], has become an extremely popular medium for sharing information including multimedia for the purpose of educating people (informal) and also during emergencies.

More recently, entertainment and sports oriented multimedia streaming applications such as Netflix [19] and online gaming applications e.g., [20], have earned immense popularity, and

S. Maharjan is with Simula Research Laboratory, Martin Linges Vei 17, Fornebu, Norway. Y. Zhang and S. Gjessing are with Simula Research Laboratory, Norway; and Department of Informatics, University of Oslo, Gaustadalleen 23, Oslo, Norway. Email: {sabita, yanzhang, steing}@simula.no. This research was partially supported by the projects 240079/F20 funded by the Research Council of Norway.

created terrific business opportunities. Such video streaming platforms have opened interfaces for crowdsourced live streaming, and have been extremely successful, especially for entertainment and sports online broadcasting, e.g., [20], [21]. This kind of trend has sparked great interest in multimedia crowdsourcing. For instance, Twitch TV [22], the world's leading video platform and community for gamers, is an example of crowdsourcing based live streaming.

Multimedia applications are resource hungry in terms of storage, processing as well as bandwidth resources. Even though mobile devices get lighter and thinner, their computational and storage capabilities can hardly keep up with users' growing demands for a media-rich experience. The emerging cloud-computing technology [23] offers new opportunities for multimedia applications and services, as it can reduce the cost of deploying and operating multimedia networks. Under the cloud-computing paradigm, system resources can be dynamically allocated to meet demands in real-time, and the users of cloud resources/services are charged according to the pay-as-you-go principle. Motivated by such considerations and developments, we propose a new cloud enabled mobile multimedia crowdsourcing framework. In such a framework, we are particularly interested in media contents generated by mobile devices (e.g., smartphones or tablets). In the context of a mobile multimedia cloud, computing resources in data centers can be instantiated into virtual machines, whose capacity can be dynamically configured for specific computation-intensive media processing tasks such as encoding, decoding, transcoding and rendering. Thereby, such tasks can be offloaded from the resource-limited mobile devices to virtual machines in the cloud [24], [25] thus making algorithms previously considered infeasible on mobile devices, feasible and practical.

Even so, multimedia crowdsourcing presents overwhelming technical challenges. For collective content production, massive server capacity is necessary to deal with online video synchronization, processing, and transcoding for highly heterogeneous video contributors and consumers. For instance, the Twitch TV [22] platform attracts over 44 million visitors per month, and every second its servers are loaded with thousands of live channels [26]. The distributed and highly dynamic sources of multimedia, as well as more stringent multimedia-QoS constraints, make the problem more challenging. In addition, heterogeneity of the multimedia sources may influence various aspects. For instance, the features and specifications of the multimedia sources, such as energy consumption rate and camera resolution are responsible for both the cost of multimedia generation and the quality of the multimedia content. What is more, the heterogeneity may also have an impact on the resources required for data acquisition, storage, processing and transmission, and consequently the quality of information that the crowdsourcer obtains. As a result, the resource requirements for the same image or for the same duration of video taken by different devices may vary. Thereby, crowdsourcing based multimedia applications indeed introduce some new and unique challenges, and essentially demand new solutions that jointly consider incentive design for the multimedia contributors, gain of the crowdsourcer and resource management for the multimedia cloud.

We model the dynamic strategies of the crowdsourcer and the contributors as a utility maximization problem by jointly considering the advantages offered by cloud based resource management, and the challenges arising from multimedia constraints and cloud capacity constraints. Such a study is of paramount importance for several reasons. First: our work is one of the first to jointly consider the multimedia constraints and the cloud resource management constraints in a crowdsourcing framework. Consequently, we attempt to reasonably model the cost incurred to the crowdsourcer for streaming the multimedia contents from the contributors, and integrate it in the crowdsourcing framework, which makes our analysis more comprehensive, and practical. Second: We also integrate heterogeneity of the contributors in terms of their cost and quality factors for the multimedia they generate. Moreover, our model is quite different from the traditional cloud based multimedia streaming frameworks, where smartphones or other devices stream the multimedia from the cloud. While our model might be slightly unconventional in this regard, we stress that our cloud enabled crowdsourcing framework demands this modified perception in the system structure of multimedia streaming.

To this end, we keep multimedia and cloud resource constraints, our cloud cost model, and the utilities of the crowdsourcer and the contributors, in quite generic forms without limiting them to specific tasks and applications. Subsequently, our model and analysis can be applicable to capture the essence of several big data based applications. For instance, apart from live streaming and sports news and updates, multimedia crowdsourcing for tasks such as forest monitoring, safer navigation of vehicles/ships, post natural catastrophe rescues etc., are some of the possible applications, which can deliver tremendous benefits to people and the society.

Our main contributions in this paper are as follows.

- We introduce a new concept of cloud-enabled multimedia crowdsourcing, and model the interactions between the crowdsourcer and the contributors as a utility maximization problem.
- We jointly consider multimedia constraints and cloud resource constraints. Moreover, we incorporate heterogeneity of the multimedia contributors to reflect the characteristic features of a real crowdsourcing scenario.
- We derive conditions (i) for the eligibility of the smartphones to participate and (ii) for the existence and uniqueness of the optimal solutions.

The remainder of the paper is organized as follows. We review the state of the art literature and place our work in the context in Section II. In Section III, we first provide a brief background on multimedia cloud, and then introduce our multimedia crowdsourcing system model. We define the utility functions for the contributors and the crowdsourcer in Section IV-A, introduce cloud resource and multimedia QoS constraints in Section IV-B, and consequently develop a cost model for the crowdsourcer, for using the cloud resources in Section IV-C. In Section IV-D, we formulate the multimedia crowdsourcing utility maximization problem, and present an algorithm to implement our scheme. To illustrate our model,

we provide numerical results and discussion in Section V. Section VI concludes the paper.

II. LITERATURE REVIEW

Conventional crowdsourcing applications have been extensively studied, e.g., [2], [3], [4], [5], [6]. With the rapid development and proliferation of advanced mobile devices such as smartphones and tablets, they have become a pervasive platform for participatory sensing and thus crowdsourcing. Consequently, several existing works have investigated applications focussing specifically on smartphones based crowdsensing, e.g., [8], [9], [10], [11].

Some studies have investigated the economic interactions among the contributors in a mobile crowdsourcing market e.g., [27], [28]. In [27] the contributors compete for a fixed reward from the crowdsourcer, by selecting the optimal number of time units to serve. In [28], the authors modeled the competitive selection process of contributors as a congestion game, maximizing the number of satisfied mobile devices, while also addressing the data redundancy issues of the crowdsourcer. In [10], the authors advocated the adoption of smartphones as a computing and networking platform for mobile crowdsourcing applications, highlighting those positive aspects that are specific to network monitoring scenarios.

An overview of various mobile crowdsensing applications is presented in [29]. The authors also discuss the unique characteristics of such applications that differentiate them from conventional sensor networks, and elaborate on the possible opportunities that such applications may provide, and illustrate the associated research challenges. The authors in [30] introduced a mobile crowdsourced sensing scenario where the contributors arrive randomly in an online manner, and proposed two online auction mechanisms that satisfy truthfulness and computational efficiency among others. In [31], the authors addressed the problem of ensuring truthfulness of the auction between the crowdsourcer(s) and the smartphones, while also integrating the dynamic nature of the smartphones.

The authors in [32] introduce a new perspective to mobile crowdsourced sensing applications, by incorporating the human mobility factor. The authors envision that human mobility offers unprecedented opportunities for both sensing, coverage and data transmission, and argue that identifying the users' contexts, mobility and social properties is important for optimizing the performance of such applications. Another work that incorporates the social aspect of the participants for mobile crowdsourcing applications is [33]. The authors in [33] exploit the social attributes of the participants, and propose a social aware and reputation management based scheme for the optimal selection of the participants. Our work in this paper differs from literature such as [28] - [33] as we introduce the concept of mobile multimedia crowdsourcing, which is a new problem that represents both great opportunities from the cloud infrastructure, and overwhelming challenges for cloud resource management and for guaranteeing multimedia QoS requirements.

One of the first studies to investigate the potentials of big data for crowdsensing and crowdsourcing is [34]. The paper

also presents constraints and challenges of creating, collecting and analyzing big data, and acting based on the analysis. In [35], the authors have presented various techniques and architectures to best utilize the cloud resources in the context of multimedia cloud computing from *media-cloud* and *cloud-media* perspectives. A thorough survey of the cloud computing based applications, related challenges and the future potentials can be found in [36]. We note that the focus in most of the related (mobile) cloud computing studies is the techniques or algorithms for cloud resource optimization. On the contrary, our main focus is to model the dynamics of the crowdsourcer and the smartphones through their economic interactions, where the cloud is an enabler.

We observe that there are few recent articles on crowdsourced live streaming e.g., [37], [26]. In [37], the authors propose a scalable hybrid cloud architecture called Gigasight that makes use of the decentralized cloud computing infrastructure (cloudlets). In [26], the authors identify the potential benefits as well as the key challenges when crowdsourced videos meet the cloud, based on real-world measurements. The authors developed a generic framework to facilitate a cost-effective cloud services for crowdsourced live streaming to accommodate geo-distributed video crowdsourcers. Our work differs in the sense that we focus on the economic interactions when multimedia crowdsourcing meets the cloud. Nevertheless, we integrate both multimedia constraints and cloud resource allocation constraints in the form of the cost of the cloud resources. In addition, as our cloud and multimedia constraints are more generic and not limited to a particular application, thus our framework and problem formulation can be useful for a wide range of applications.

III. PRELIMINARIES AND SYSTEM MODEL

A. The Multimedia Cloud: Background

The cloud is an integral component of our multimedia crowdsourcing framework. In a general purpose cloud, a utility-like mechanism is deployed to allocate the computing/processing and storage resources, which is effective for general data services. However, for multimedia applications, simultaneous bursts of multimedia data access, processing, and transmissions in the cloud may create a bottleneck in a general-purpose cloud. Therefore, using a general-purpose cloud to deal with multimedia services may suffer from unacceptable media quality of service (QoS). The main constraints come from the stringent requirements on the multimedia streaming, such as end-to-end latency and jitter requirements, packet loss requirements, and buffering and streaming speed, which may be greatly affected by network heterogeneity and device heterogeneity.

Multimedia cloud service providers pull together a pool of shared ICT resources, including processing, storage, and bandwidth resources, and allocate them elastically to various media-related tasks according to their real-time application demands. Computing capacity could come from a diverse set of resources, for instance, data centers that house a fleet of general-purpose blade servers and CPU/GPU arrays that are dedicated for multimedia processing tasks like image

segmentation, feature extraction and video coding [38], [39]. These computation facilities can utilize super-sized storage distributed across different locations that can be requested on demand [35].

In our framework, the multimedia cloud is an enabler of the multimedia crowdsourcing paradigm. We consider that the cloud implements efficient techniques and algorithms for allocating and reconfiguring the cloud resources in an elastic manner to complete tasks in multimedia networks such as multimedia processing, distribution, rendering and analytics, among others, and is not directly involved as a player in the incentive design process. Instead, our major focus in this paper is on the economic interactions among the contributors, and between them and the crowdsourcer.

B. Multimedia Crowdsourcing: System Model

Fig. 1 depicts the big picture of our framework. The multimedia cloud houses resources such as huge amount of storage and powerful processing units like CPU and GPU. The crowdsourcer streams the multimedia information through the cloud. Fig. 2 illustrates the interactions between the crowdsourcer and the contributors. The crowdsourcer broadcasts the task description and its requirements e.g., the minimum required resolution of the camera. Based on the eligible contributors, the crowdsourcer computes and announces a total reward for them. Correspondingly, the contributors compute their optimal duration of participation i.e., the time duration for multimedia generation. When, the contributors upload the multimedia contents to the cloud, the crowdsourcer pays them for their service, and streams the contents.

The overall objective of this framework is to maximize the utility of both the crowdsourcer and the contributors. Therefore, both crowdsourcers and contributors need to adapt their behaviors in order to approach their respective goals. The crowdsourcer chooses an optimal reward to maximize its own utility, taking into consideration the outcome of the competition among the contributors. The contributors also choose their optimal strategies according to the reward announced by the crowdsourcer.

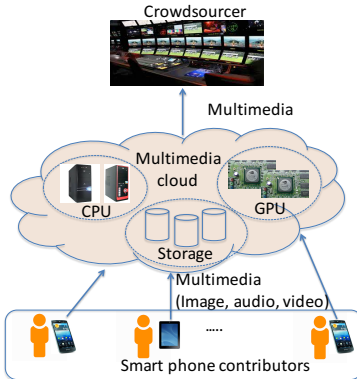


Fig. 1. A typical representation of a cloud-enabled multimedia crowdsourcing framework

There are several characteristic features of our framework and model that we would like to emphasize.

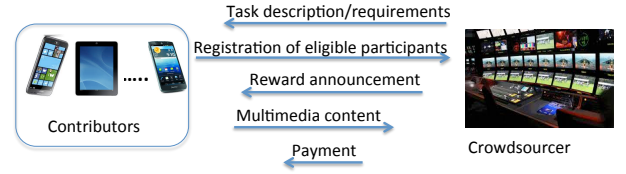


Fig. 2. Communication between the mobile devices and the crowdsourcer

First, in our model, the smartphones are the creators of the multimedia content, and the crowdsourcer is the end-user. In this regard, our model is quite different from the traditional cloud based multimedia streaming frameworks such as in [26] and [40], where smartphones or other devices stream the multimedia from dedicated servers that have huge storage and powerful processing resources. While our model might look rather unconventional, we stress that our cloud enabled crowdsourcing framework demands this modified perception in the system structure of multimedia streaming. Thus our system model is practical, and is expected to be useful in a wide range of multimedia crowdsourcing applications. Second, apart from the constraints dictated by multimedia storage, processing and transmission requirements, we incorporate heterogeneity of the contributors, specifically tailored to multimedia crowdsourcing oriented applications, which plays an important role in the strategies of both the crowdsourcer and the contributors.

IV. OPTIMAL INCENTIVE MECHANISM: PROBLEM FORMULATION

We consider the set of smartphones $\mathcal{N} := \{1, 2, \dots, j, \dots, N\}$ interested in creating multimedia content for the crowdsourcer. The interactions between the crowdsourcer and the mobile devices essentially consists of two stages: In the first stage, the crowdsourcer announces its strategy, i.e., the total reward $R > 0$ for the contributors; in the second stage, each contributor chooses the duration of its service, i.e., t_j for smart phone j , in order to maximize its own payoff.

Let q_j represent the value of the video taken by contributor j to the crowdsourcer. Clearly, q_j depends on the camera resolution and other features or settings, which we assume, is given, for a particular smartphone. For instance, q_j may represent a combination of the resolution, frame rate and/or shutter speed. The cost for contributor j is $k_j t_j$, where k_j is the unit cost factor, that may reflect among others, the energy consumption rate or the battery depletion rate of the smartphone. Note that the total time of service from a contributor can be just for one task, or it can be for multiple tasks at different times.

A. Utility Functions

As q_j is given, it is reasonable to consider that the reward received by contributor j is proportional to t_j . Then the utility of user j is the difference between the reward it gets and the cost for creating the multimedia content, i.e.,

$$u_j := \frac{q_j t_j}{\sum_{i \in \mathcal{N}} q_i t_i} R - t_j k_j. \quad (1)$$

The utility function (1) can be related to the definition of a contributor's utility in [27], thus ensuring that our definition of the terms are inline with a widely accepted practice. Nonetheless, in [27], the authors design incentive mechanism for smartphone crowdsensing, while our scenario as well as our formulation are driven by the multimedia contents, the cloud constraints and the joint modelling.

As each user seeks to maximize its own utility, it should satisfy the Individual Rationality (IR) property, i.e., the multimedia crowdsourcing scheme should ensure that each contributor can obtain a non-negative utility, which will be the case when $R \geq \frac{k_j}{q_j} \sum_{i \in \mathcal{N} \setminus j} q_i t_i, \forall j \in \mathcal{N}$. Thus the user-side optimization problem can be formulated for user j as

$$\begin{aligned} & \max_{t_j} u_j(t_j) \\ \text{s.t. } & R \geq \frac{k_j}{q_j} \sum_{i \in \mathcal{N} \setminus j} q_i t_i. \end{aligned} \quad (2)$$

We deploy the following utility function for the crowdsourcer

$$U_{CS} = \alpha \ln \left(1 + \sum_{j \in \mathcal{N}} \ln(1 + q_j t_j) \right) - R, \quad (3)$$

where $\alpha > 1$ is a parameter specific to the crowdsourcer. The $\ln(1 + q_j t_j)$ term reflects the crowdsourcer's diminishing return on the service of contributor j , and the outer log term reflects the crowdsourcer's diminishing return on the number of participants. This kind of utility function has been widely accepted to represent the payoff of a crowdsourcer, e.g., [27].

B. Media and Cloud constraints

While the mobile devices are the sources of the multimedia, the multimedia is stored, processed and is transmitted through the cloud infrastructure. Let r, σ represent the link bandwidth and the total storage and processing resources, respectively, required by the cloud for storage, processing and transmission of the multimedia from the smartphones to the crowdsourcer. Let $r_{CS, \max}, \sigma_{CS, \max}$ be the maximum bandwidth and storage resources the cloud can leverage. Then, the capacity constraints for the cloud resources can be expressed in a relatively generic form as

$$\begin{cases} r \leq r_{CS, \max}, \\ \sigma \leq \sigma_{CS, \max}. \end{cases} \quad (4)$$

To formulate the QoS, it is important to figure out a suitable QoS metric for multimedia crowdsourcing based applications, which certainly depends on the particular application and its requirements.

Usually, for multimedia, the metrics include end-to-end latency and/or jitter constraints, and reliability requirements such as packet drop/loss rate, so that the quality of presentation is acceptable under diverse network conditions and resource constraints. Specifically, for video, the QoS metrics such as end-to-end latency and jitter are important. For real time and online streaming applications, the requirement on the end-to-end latency might be strict. When humans interact with video in a live video conference or when playing a game, latency

lower than 100 ms is considered acceptable [41]. For multimedia crowdsourcing based (almost) real-time surveillance and monitoring applications, the end-to-end latency within a few seconds is expected to be tolerable. Jitter is an important measure of video content, as it mainly affects the quality of presentation [42]. Even for images, jitter can be an important quality of experience measure especially for big data oriented monitoring and control applications such as forest monitoring.

We therefore consider that the most stringent QoS constraint for multimedia crowdsourcing is related to jitter, which can be expressed as

$$\Delta_j \leq \Delta_{\max}, \forall j \in \mathcal{N}, \quad (5)$$

where Δ_j, Δ_{\max} are the jitter in the multimedia from contributor j , and the maximum allowed jitter threshold as required by the crowdsourcer for a particular application, respectively. The objective of the crowdsourcer can therefore be expressed as

$$\begin{aligned} & \max_R U_{CS} = \alpha \ln \left(1 + \sum_{j \in \mathcal{N}} \ln(1 + q_j t_j) \right) - R : \\ \text{s.t. } & (4), (5). \end{aligned} \quad (6)$$

C. Cost of the Cloud Resources

Complex optimization methods and algorithms are deployed to optimal provisioning of the cloud resources. Nonetheless, for the end-users the cloud is simply a service provider, providing its infrastructure as a service for dynamic and elastic resource allocation to meet their demands on a *pay-as-you-go* principle. The resource optimization techniques and algorithms used for cloud resource management are not transparent to the end-users. As a result, from the end-user's perspective, which is the crowdsourcer in our framework, what is important, is the cost function that governs the cost that it has to pay for using the cloud infrastructure/resources as a service. Defining a function for the cloud cost $C_{cl}(\cdot)$ is a complex task. Interestingly, however, the cost function can be characterised with some generic features that hold the essential elements of charging the end-users, which will be useful in modelling our cloud-enabled multimedia crowdsourcing framework.

From the cloud perspective, the usage of the cloud resources should be within the maximum capacity that the cloud infrastructure can provide to ensure the capacity constraints such as (4), while still meeting the QoS requirements e.g., (5). Mathematically, the cloud can deploy two alternative formulations for resource management to meet these constraints [36]:

- 1) bounded capacity constraint, in which the resource usage is strictly less than a predetermined threshold, and
- 2) penalized capacity constraint, in which the resource usage can be larger than a predetermined threshold, but a penalty function is associated with the over-provisioned capacity.

We therefore, consider that the cloud resources can ensure that the constraints (4) and (5) can be met, at a certain cost. The utility of the crowdsourcer (3) can thus be modified as

$$U_{CS} = \alpha \ln \left(1 + \sum_{j \in \mathcal{N}} \ln(1 + q_j t_j) \right) - (R + \beta C_{cl}), \quad (7)$$

where β is a scaling factor.

Let us take a look at some characteristic details from the perspective of cloud resource management for multimedia. The cloud deployment cost consists of two major components:

- 1) server cost due to the total storage and processing capability of a server; and
- 2) link cost due to the bandwidth reserved and data transmitted between pairs of servers.

For efficient cloud resource management, if the storage cost is relatively low, for instance, then increasing the storage capacity may lead to lower overall cost. Similarly, if the link bandwidth cost is relatively low, server collaboration and hence purchasing more link capacity may lead to cheaper deployment cost.

Let E_m, P_m, B_m denote the total storage capacity and the processing capacity of server m , and the total bit rate that server m can serve other servers, respectively. Let us represent the link bandwidth reserved for the directed traffic from server m to server n by r_{mn} , and the actual data transferred through the link by D_{mn} . For any remote server $m \in \mathcal{S}$, the cost of a server $c_{S,m}$, and the link cost due to the directed traffic from server m to server n , i.e., $c_{L,mn}$ can be represented as

$$\begin{cases} c_{S,m} = f_{S,m}^c(B_m, P_m), \forall m \in \mathcal{S}, \\ c_{L,mn} = f_{L,mn}^c(D_{mn}, r_{mn}), \forall m, n \in \mathcal{S}. \end{cases} \quad (8)$$

where $f_{S,m}^c$ is a monotonically non-decreasing function in E_m, P_m and B_m , and $f_{L,mn}^c$ is a monotonically non-decreasing function in r_{mn} and/or D_{mn} . Let $d_{S,m}, d_{L,mn}$ be the delay due to the upload streaming of server m to support video requests from other servers, and the delay due to the directed traffic from server m to server n , respectively. Then, $d_{S,m}, d_{L,mn}$ can be written as

$$\begin{cases} d_{S,m} = f_{S,m}^d(B_m, P_m), \forall m \in \mathcal{S}, \\ d_{L,mn} = f_{L,mn}^d(D_{mn}, r_{mn}), \forall m, n \in \mathcal{S}. \end{cases} \quad (9)$$

where $f_{S,m}^d$ is a monotonically non-decreasing function in B_m and monotonically non-increasing function in P_m , and $f_{L,mn}^d$ is a monotonically non-decreasing function in P_{mn} and monotonically non-increasing function in r_{mn} . Let D be the upper bound of the total delay to retrieve the multimedia from other servers. Then, to meet the QoS requirement (5), it is required that

$$d_{S,m} + d_{L,mn} \leq D, \forall m, n \in \mathcal{S}. \quad (10)$$

The total cloud deployment cost C_{cl} can be expressed as

$$C_{cl}(\cdot) = \sum_{m \in \mathcal{S}} c_{S,m} + \sum_{m,n \in \mathcal{S}} c_{L,mn}. \quad (11)$$

For a given topology, delay upper bound, video streaming demand, the cost functions $f_{S,m}^c, f_{L,mn}^c$ and the delay functions

$f_{S,m}^d, f_{L,mn}^d$, resources are dynamically and elastically assigned in the cloud to minimize the total deployment cost. From the crowdsourcer's perspective, given the cost functions, delay functions, jitter requirements, and the cloud resource allocation techniques and algorithms, the cloud cost depends on the amount of multimedia the end-user streams i.e., $\sum_{i \in \mathcal{N}_s} q_i t_i$. To this end, $C_{cl}(\cdot)$ can be more specifically represented as a continuous non-decreasing function of $\sum_{i \in \mathcal{N}_s} q_i t_i$ in our model. Now, the utility of the crowdsourcer (7) can be expressed as

$$U_{CS} = \alpha \ln \left(1 + \sum_{i \in \mathcal{N}_s} \ln(1 + q_i t_i) \right) - R - \beta C_{cl} \left(\sum_{i \in \mathcal{N}_s} q_i t_i \right). \quad (12)$$

D. Crowdsourcing Utility Maximization: Problem Formulation

For given q_j for contributor j , differentiating (1) w.r.t. t_j , we get

$$\frac{\partial u_j}{\partial t_j} = \frac{q_j R}{\sum_{i \in \mathcal{N}} q_i t_i} \left(-\frac{q_j t_j}{\sum_{i \in \mathcal{N}} q_i t_i} + 1 \right) - k_j \quad (13)$$

Differentiating (13) further w.r.t. t_j , we obtain

$$\frac{\partial^2 u_j}{\partial t_j^2} = -\frac{2q_j^2 R \sum_{i \in \mathcal{N} \setminus j} q_i t_i}{(\sum_{i \in \mathcal{N}} q_i t_i)^3}. \quad (14)$$

(14) implies that $\frac{\partial^2 u_j}{\partial t_j^2} < 0$, i.e., the utilities of the contributors are strictly concave in t_j , i.e., for any reward $R > 0$ and for given strategies of other users \mathbf{t}_{-j} , the optimal response strategy of user j is unique if it exists. Clearly, we are interested only in the cases when $\sum_{i \in \mathcal{N} \setminus j} t_i > 0$. Now, using $\frac{\partial u_j}{\partial t_j} = 0$, and after simplification we can obtain the only possible solution for t_j as

$$t_j = -\frac{\sum_{i \in \mathcal{N} \setminus j} q_i t_i}{q_j} + \sqrt{\frac{R \sum_{i \in \mathcal{N} \setminus j} q_i t_i}{k_j q_j}}. \quad (15)$$

From (15), it is clear that $t_j > 0$ if

$$R > \frac{k_j}{q_j} \sum_{i \in \mathcal{N} \setminus j} q_i t_i, \forall j \in \mathcal{N}, \quad (16)$$

which is the IR constraint defined in (2). Thus the optimal solution for smartphone j , i.e., t_j^* for given R can be obtained as

$$t_j^* = \begin{cases} -\frac{\sum_{i \in \mathcal{N} \setminus j} q_i t_i}{q_j} + \sqrt{\frac{R \sum_{i \in \mathcal{N} \setminus j} q_i t_i}{k_j q_j}}, \\ \text{if } R > \frac{k_j}{q_j} \sum_{i \in \mathcal{N} \setminus j} q_i t_i, \forall j \in \mathcal{N}, \\ 0 \text{ Otherwise.} \end{cases} \quad (17)$$

Clearly, only those smartphones that satisfy (16) can participate in the crowdsourcing incentive design process. Now, let us represent the set of the eligible smartphones, as \mathcal{N}_s , such that $\mathcal{N}_s \subseteq \mathcal{N}$, and $N_s = |\mathcal{N}_s|$. For the contributors, we can write t_j as

$$t_j = -\frac{\sum_{i \in \mathcal{N}_s \setminus j} q_i t_i}{q_j} + \sqrt{\frac{R \sum_{i \in \mathcal{N}_s \setminus j} q_i t_i}{k_j q_j}}.$$

Moving the first term from the RHS to the LHS gives

$$t_j + \frac{\sum_{i \in \mathcal{N}_s \setminus j} q_i t_i}{q_j} = \sqrt{\frac{R \sum_{i \in \mathcal{N}_s \setminus j} q_i t_i}{k_j q_j}}.$$

After simplifying the above equation, we get

$$\frac{\sum_{i \in \mathcal{N}_s} q_i t_i}{q_j} = \sqrt{\frac{R \sum_{i \in \mathcal{N}_s \setminus j} q_i t_i}{k_j q_j}}.$$

Squaring both sides and after simplification we can obtain

$$\left(\sum_{i \in \mathcal{N}_s} q_i t_i \right)^2 = R \frac{q_j}{k_j} \sum_{i \in \mathcal{N}_s \setminus j} q_i t_i, \forall i, j \in \mathcal{N}_s. \quad (18)$$

Now equating the RHSs of (17) for $j = 1, 2, \dots, N_s$, i.e., using $\frac{q_1}{k_1} \sum_{i \in \mathcal{N}_s \setminus 1} q_i t_i = \frac{q_2}{k_2} \sum_{i \in \mathcal{N}_s \setminus 2} q_i t_i = \dots = \frac{q_{N_s}}{k_{N_s}} \sum_{i \in \mathcal{N}_s \setminus N_s} q_i t_i$, $t_i, \forall i \in \mathcal{N}_s \setminus j$ can be expressed in terms of t_j . Then, substituting $t_i, \forall i \in \mathcal{N}_s = f(t_j)$ in (17), we can obtain t_j in the closed form for given R , as

$$t_j = \frac{(N_s - 1)R}{\left(\sum_{i \in \mathcal{N}_s} k_i \prod_{m \in \mathcal{N}_s \setminus i} q_m \right)^2} \left(\sum_{i \in \mathcal{N}_s} k_i \prod_{m \in \mathcal{N}_s \setminus i} q_m - (N_s - 1)k_j \prod_{m \in \mathcal{N}_s \setminus j} q_m \right). \quad (19)$$

$\forall j \in \mathcal{N}_s$. From (19), it can be seen that (16), can be equivalently expressed in a closed form as

$$k_j < \frac{\sum_{i \in \mathcal{N}_s \setminus j} k_i \prod_{m \in \mathcal{N}_s \setminus i} q_m}{(N_s - 2) \prod_{m \in \mathcal{N}_s \setminus j} q_m}, \forall j \in \mathcal{N}, \quad (20)$$

which implies that a smartphone can participate in the multimedia crowdsourcing process if its cost factor k_j is less than a certain threshold, advised by the RHS of (20).

Remark 1. As the contributors are interested in higher utilities, a concern about their truthfulness is natural. In our model however, the contributors do not actively bid. Instead, they register or report their parameters such as q_j and k_j to the crowdsourcer. Interestingly, in our model, the crowdsourcer pays to the contributors only after streaming the multimedia content from them, which eliminates the possibility of the contributors registering false values of q_j . Regarding k_j , it may appear that the contributors can cheat by registering a higher value. This however, will not be the case. If, k_j increases, there is a risk of violating (20), which will simply render the user ineligible to participate. As each interested user has the information only about its own parameters and does not know k_i, k_j, q_m, N_s to compute the RHS of (20), it is not possible for a user to report a higher value of k_j while still satisfying (20). Similarly, if the contributor registers its k_j smaller than the actual value, it will end up serving longer while getting paid less, which is certainly not desirable to the contributors. Thus, the best in the interest of the contributors is to register the true values of their parameters.

We defined a cost function for the use of the cloud resources in Section IV-C. As our assumptions on deriving the cost function $C_{cl}(\sum_{i \in \mathcal{N}_s} q_i t_i)$ are quite generic, it can be highly relevant for general multimedia crowdsourcing based frameworks. Consequently, we believe that such a generic model can

be employed for many cloud enabled applications and services. This establishment can be well related to the results in the literature. For instance, the simulation results in [35], obtained by running parallel photosynthing code in an HPC cluster with nine servers, demonstrate that the time taken for tasks such as image conversion, feature extraction and image matching, all decrease proportionally with increased number of servers, for the cases of both 200 and 400 images. The gain in the required time to execute the image processing tasks certainly come from the additional server cost, which can be associated with the server cost $c_{S,m}$ in (8). On a similar note, the link costs offered by the Google Cloud [43], remarkably suggest that the costs charged to the users increase as the amount of data used per month increases. Such data further supports the relevance of employing a continuous non-decreasing function w.r.t. the total amount of data to represent the cost for the use of the cloud resources. Some relevant examples of the function $C_{cl}(\cdot)$ can be logarithmic: $\ln(1 + \sum_{i \in \mathcal{N}_s} q_i t_i)$, sigmoidal: $\frac{\sum_{i \in \mathcal{N}_s} q_i t_i}{\sqrt{1 + (\sum_{i \in \mathcal{N}_s} q_i t_i)^2}}$ for $\sum_{i \in \mathcal{N}_s} q_i t_i \geq 0$, and even linear: $\gamma_0 + \gamma (\sum_{i \in \mathcal{N}_s} q_i t_i)$, where γ, γ_0 are the slope and the intercepts, respectively, for the linear cost function. For the bounded capacity constraint based formulations, $C_{cl}(\cdot)$ can be considered at most linear in $\sum_{i \in \mathcal{N}_s} q_i t_i$. For the penalized capacity constraint based formulations, $C_{cl}(\cdot)$ can be of higher order in $\sum_{i \in \mathcal{N}_s} q_i t_i$, such as quadratic or cubic for the overprovisioned capacity.

Theorem 1. *Unique optimal solutions exist for both the duration of service from the contributors, and the reward to the contributors from the crowdsourcer, provided*

$$\alpha \left(\sum_{i \in \mathcal{N}_s} \left(\frac{q_i S_i}{1 + q_i S_i R} \right)^2 T + \left(\sum_{i \in \mathcal{N}_s} \frac{q_i S_i}{1 + q_i S_i R} \right)^2 \right) > \beta T^2 C_{cl}''(R \sum_{i \in \mathcal{N}_s} q_i S_i) \quad (21)$$

Proof. Substituting t_j from (19) in (12), we get.

$$U_{CS} = \alpha \ln(1 + \sum_{i \in \mathcal{N}_s} \ln(1 + q_i S_i R)) - R - \beta C_{cl}(R \sum_{i \in \mathcal{N}_s} q_i S_i). \quad (22)$$

where

$$S_i = \frac{N_s - 1}{\left(\sum_{i \in \mathcal{N}_s} k_i \prod_{m \in \mathcal{N}_s \setminus i} q_m \right)^2} \left(\sum_{i \in \mathcal{N}_s} k_i \prod_{m \in \mathcal{N}_s \setminus i} q_m - (N_s - 1)k_j \prod_{m \in \mathcal{N}_s \setminus j} q_m \right). \quad (23)$$

Taking the first first derivative of (12), we get

$$\frac{\partial U_{CS}}{\partial R} = \frac{\alpha}{T} \left(\sum_{i \in \mathcal{N}_s} \frac{q_i S_i}{1 + q_i S_i R} \right) - 1 - \beta C_{cl}' \left(R \sum_{i \in \mathcal{N}_s} q_i S_i \right). \quad (24)$$

where $T = 1 + \sum_{i \in \mathcal{N}_s} \ln(1 + q_i S_i R)$. Differentiating (24) further w.r.t. R gives

$$\frac{\partial^2 U_{CS}}{\partial R^2} = -\beta C_{cl}''(R \sum_{i \in \mathcal{N}_s} q_i S_i) - \frac{\alpha}{T^2} \left(\sum_{i \in \mathcal{N}_s} \left(\frac{q_i S_i}{1 + q_i S_i R} \right)^2 T + \left(\sum_{i \in \mathcal{N}_s} \frac{q_i S_i}{1 + q_i S_i R} \right)^2 \right) \quad (25)$$

Since $C_{cl}(R \sum_{i \in \mathcal{N}_s} q_i S_i)$ is continuous and non-decreasing, $C_{cl}'(R \sum_{i \in \mathcal{N}_s} q_i S_i) \geq 0$ for $R > 0$, and

$C_{cl}''(R \sum_{i \in \mathcal{N}_s} q_i S_i) \leq 0$ for $R > 0$. Interestingly, from (25), we can see that the utility of the crowdsourcer defined in (7) is a strictly concave function of R if (21) is true.

Hence a unique optimal solution exists for R when (21) is true, which can be obtained by numerically solving $\frac{\partial U_{CS}}{\partial R} = 0$. Earlier, we have established that for $R > 0$, a unique optimal solution exists for the competition among the contributors given by (19). The optimal solutions for our crowdsourcing scheme are therefore

$$[R^*, \{t_j^* \forall j \in \mathcal{N}_s\}] := \left[[R]_{\frac{\partial U_{CS}}{\partial R}=0}, (19) \forall j \in \mathcal{N}_s \right]. \quad (26)$$

□

In order to gain further insight into the cloud cost let us consider a linear and a logarithmic cost function, which are strictly increasing, and non-decreasing, respectively.

1) *Case 1:* $C_{cl}(\cdot) = \gamma(R \sum_{i \in \mathcal{N}_s} q_i S_i) + \gamma_0$. In this case, (24) and (25) take the forms,

$$\frac{\partial U_{CS}}{\partial R} = \frac{\alpha}{T} \left(\sum_{i \in \mathcal{N}_s} \frac{q_i S_i}{1 + q_i S_i R} \right) - 1 - \beta \gamma \sum_{i \in \mathcal{N}_s} q_i S_i, \quad (27)$$

and

$$\frac{\partial^2 U_{CS}}{\partial R^2} = -\frac{\alpha}{T^2} \left(\sum_{i \in \mathcal{N}_s} \left(\frac{q_i S_i}{1 + q_i S_i R} \right)^2 T + \left(\sum_{i \in \mathcal{N}_s} \frac{q_i S_i}{1 + q_i S_i R} \right)^2 \right) \quad (28)$$

respectively. From (28) it is clear that a unique optimal solution for R exists when the cloud cost $C_{cl}(\cdot)$ is linear in $\sum_{i \in \mathcal{N}_s} q_i t_i$. A closer look at (12), (27) and (28) reveals that a unique optimal solution always exists for $C_{cl}(\cdot)$ of order ≥ 1 , a special case for which (21) is always true since $C_{cl}''(R \sum_{i \in \mathcal{N}_s} q_i S_i) = 0$.

2) *Case 2:* $C_{cl}(\cdot) = \ln(1 + R \sum_{i \in \mathcal{N}_s} q_i S_i)$: In this case, taking the first and the second order partial derivatives of (12) w.r.t. R , we can obtain

$$\frac{\partial U_{CS}}{\partial R} = \frac{\alpha}{T} \left(\sum_{i \in \mathcal{N}_s} \frac{q_i S_i}{1 + q_i S_i R} \right) - 1 - \beta \frac{\sum_{i \in \mathcal{N}_s} q_i S_i}{1 + R \sum_{i \in \mathcal{N}_s} q_i S_i}, \quad (29)$$

and

$$\frac{\partial^2 U_{CS}}{\partial R^2} = -\frac{\alpha}{T} \sum_{i \in \mathcal{N}_s} \left(\frac{q_i S_i}{1 + q_i S_i R} \right)^2 - \frac{\alpha}{T^2} \left(\sum_{i \in \mathcal{N}_s} \frac{q_i S_i}{1 + q_i S_i R} \right)^2 + \beta \left(\frac{\sum_{i \in \mathcal{N}_s} q_i S_i}{1 + R \sum_{i \in \mathcal{N}_s} q_i S_i} \right)^2, \quad (30)$$

respectively. From (30), we can see that, $\frac{\partial^2 U_{CS}}{\partial R^2} < 0$ only if

$$\alpha \left(\sum_{i \in \mathcal{N}_s} \frac{(q_i S_i)^2}{(1 + q_i S_i R)^2} T + \left(\sum_{i \in \mathcal{N}_s} \frac{q_i S_i}{1 + q_i S_i R} \right)^2 \right) > \beta T^2 \left(\frac{\sum_{i \in \mathcal{N}_s} q_i S_i}{1 + R \sum_{i \in \mathcal{N}_s} q_i S_i} \right)^2, \quad (31)$$

which is (21). Other functions such as sigmoid: $\frac{\sum_{i \in \mathcal{N}_s} q_i t_i}{\sqrt{1 + (\sum_{i \in \mathcal{N}_s} q_i t_i)^2}}$ for $\sum_{i \in \mathcal{N}_s} q_i t_i \geq 0$ need similar condition as (31) to be true for the existence of the unique optimal solution for the crowdsourcer's optimization problem (7). This example makes us argue that such a statement is true for any cloud cost function $C_{cl}(\cdot)$ of order less than linear.

3) *Implementation:* The crowdourcer and the smartphone participants interact with each other through an interface such as an app or a website. The crowdsourcer will publicly announce the need for the service, task description and its requirements for the participants through the interface. When the interested participants try to register, only those eligible according to the q_i requirement are selected in the first filtering round, forming the set \mathcal{N} . Next, from the interested participants in the set \mathcal{N} , only those that satisfy (20), make it through as contributors, and form the subset \mathcal{N}_s . Then, for the set $\{S_1, S_2, \dots, S_{N_s}\}$ computed from (23), and for the given cloud cost function $C_{cl}(\cdot)$, the crowdsourcer computes its optimal reward for the contributors R^* , and correspondingly, the contributors create the multimedia content for the duration $t_j^* = S_j R^*$. The skeleton of the implementation is shown in Algorithm 1.

Algorithm 1 : Algorithm for implementing our cloud multimedia crowdsourcing scheme

```

1: From the interested participants, choose a subset  $\mathcal{N}$  with the
   participants that satisfy the  $q_i$  requirement from the crowdsourcer.
2: For  $j = 1, 2, 3, \dots, N$ : For given  $q_1, q_2, \dots, q_N$  and
    $k_1, k_2, \dots, k_N$ 
3:   If (20) is true,
4:     participant  $j$  is eligible, i.e.,  $j \in \mathcal{N}_s$ 
5:   end
6: end
7: For  $j = 1, 2, 3, \dots, N_s$ : For given  $q_1, q_2, \dots, q_{N_s}$  and
    $k_1, k_2, \dots, k_{N_s}$ 
8:   Compute  $S_j$  using (23)
9: end
10: Find  $R^*$  using (24)= 0 and announce it to the participants.
11: For  $j = 1, 2, 3, \dots, N_s$ 
12:   Compute  $t_j^*$  using  $t_j^* = S_j R^*$ .
13: end
14: Create the multimedia content for the crowdsourcer for the
   duration  $t_j^*$ , and upload it through the interface.
15: Crowdsourcer: Pay to the contributors and stream the multimedia
   contents.
```

Now, to illustrate the overhead of Algorithm 1, let us analyze its runtime complexity. Without loss of generality, we consider that the time required for performing a mathematical operation is one unit for addition, multiplication or comparison or assignment. As the task descriptions and the q_i requirements are publicly announced by the crowdsourcer through the interface, the interested candidates that do not satisfy the q_i requirement, will be filtered out in Step 1. The **If** loop in lines 3 – 5 need a runtime of order $O(1)$, and with the **For** loop in lines 2 – 6, the time complexity is of order $O(N)$. The **For** loop in lines 7 – 9, incur a complexity of order $O(N_s)$, where $N_s \leq N$. For simplicity, we consider that solving (24)= 0 to find R^* also takes one unit of time, which is not precise but still a safe assumption as Step 10 is a one-time operation for the whole crowdsourcing process. The **For** loop in lines 11 – 13 needs $O(N_s)$ time. Since $N \geq N_s$, the overhead of Algorithm 1 in terms of runtime can be expressed as $O(N)$, which makes Algorithm 1 of reasonably low overhead and

thus practical.

Remark 2. Note that in our illustration of the runtime overhead of Algorithm 1 above, we have considered the runtime for the operations like comparison in (20), the assignment in (23) and for numerically solving (24) = 0 as one unit each. The actual number of operations each of them involves depends on N, N_s , the specific numerical method used for solving (24) = 0 and the number of iterations incurred. In this regard, our definition of the runtime is relatively coarse. It is a however a common practice to adopt such definitions according to the number of tasks or the program execution time.

V. ILLUSTRATIVE NUMERICAL RESULTS

In this section, we present the numerical results portraying the performance of our multimedia crowdsourcing framework and discuss them. The parameters used for the purpose of illustration are as follows, unless otherwise mentioned. k_j, q_j are assigned random values from the interval (0 2), and (0.8 1.2), respectively. α, γ, γ_0 are set to 35, 0.5, 0, respectively. N_s is varied from 10 to 500. β is 0.5 and 10, for the linear and the logarithmic cloud cost functions, respectively.

In Figs. 3-5, we illustrate the performance of our crowdsourcing multimedia framework with respect to (w.r.t.) the number of contributors.

Fig. 3 shows that as the number of eligible contributors to perform the task(s) assigned by the crowdsourcer increases from 10 to about 200, the crowdsourcer is encouraged to invest more in them. When the number of the interested and eligible smartphones rises further, however, the crowdsourcer turns relatively conservative in raising the reward. There are two main factors behind such a strategy of the crowdsourcer. First, the increment in the value of the information decreases if the number of the smartphones willing to participate is too high. Second, the cloud cost $C_{cl}(\cdot)$ and the reward for the smartphones R may suppress the small gain obtained from the additional contributors. The reward strategies considering both the linear and the logarithmic cost of the cloud resources, follows similar trends, but the reward is considerably less compared to the case when the cloud resource costs are not directly included in the formulation.

We illustrate the average duration of participation from a contributor in Fig. 4. We observe that an individual smartphone contributes less as the number of participants increases. Such a behavior stems from the fact that the contribution share of each smartphone is likely to be valued less as the number of participants willing to provide the same/similar service increases.

Fig. 5 suggests that the utility of the crowdsourcer improves decently up to a certain number of participants, and then the increase in the utility becomes less visible beyond that. The payoff from the increased number of participants is dominant for the crowdsourcer until about $N_s = 200$. After this point, the crowdsourcer does not respond to the increase in N_s with a proportionally higher reward (Fig. 3). The schemes that incorporate the cost of the cloud resources, naturally demonstrate diminished returns compared to the model without considering the cloud costs.

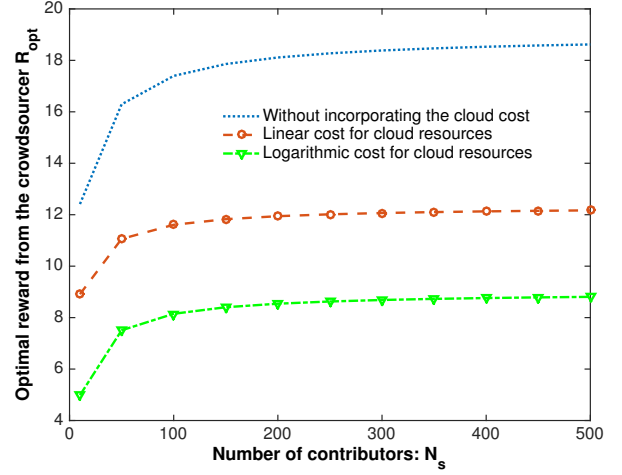


Fig. 3. Optimal reward w.r.t. the number of contributors

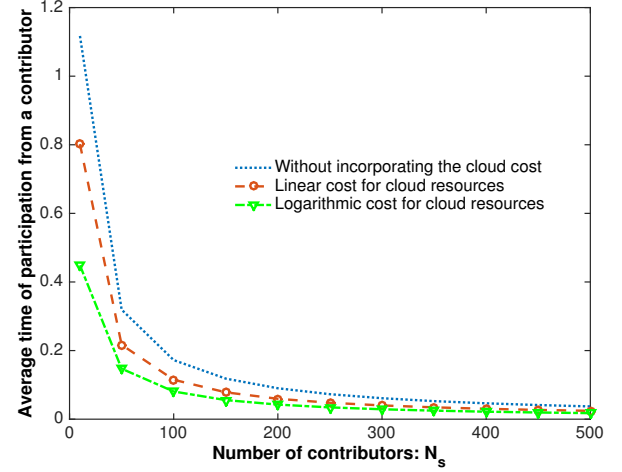


Fig. 4. Average service time of the contributors w.r.t. their number

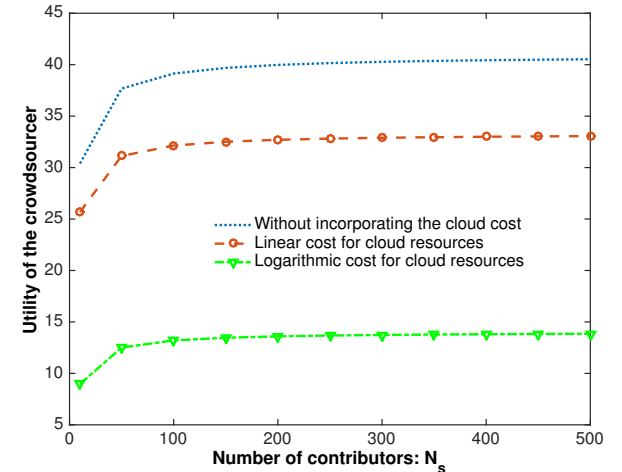


Fig. 5. Utility of the crowdsourcer w.r.t. the number of contributors N_s

Figs. 6-8, present the performances w.r.t. the average cost of participating for the contributors.

Fig. 6 implies that the crowdsourcer must increase the reward R if the average cost of participating increases as it is necessary to incentivize the smartphone participants to contribute even when their costs are higher. Even so, the crowdsourcer leverages at least about a one-third less reward

for a logarithmic cloud cost function, and further less for a linear cloud cost function, in our scheme compared to the case without considering the cloud cost. Fig. 7 shows that the contributors respond by reduced duration towards the increased cost.

Fig. 8 suggests that the utility of the crowdsourcer degrades as the cost of the contributors increases. The increase in the cost of participating leads to smaller duration of contribution from the smartphones (Fig. 7). Moreover, the crowdsourcer increases the total reward to the smartphones responding to the increase in their costs (Fig. 6). Both of these factors contribute in shrinking the utility of the crowdsourcer. The schemes that incorporate the cost of the cloud resources, naturally demonstrate diminished returns compared to the model without considering the cloud costs.

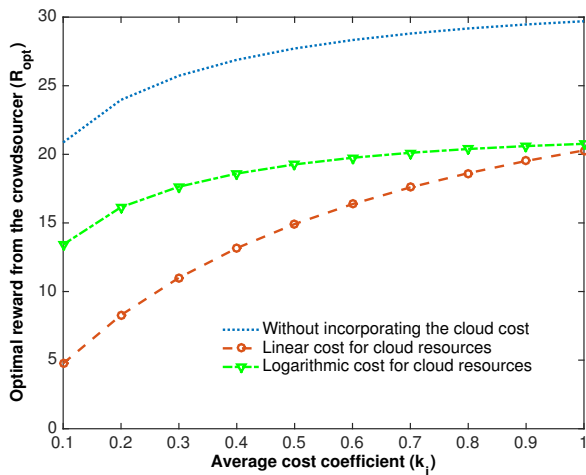


Fig. 6. Optimal reward R from the crowdsourcer w.r.t. average cost coefficient k_j

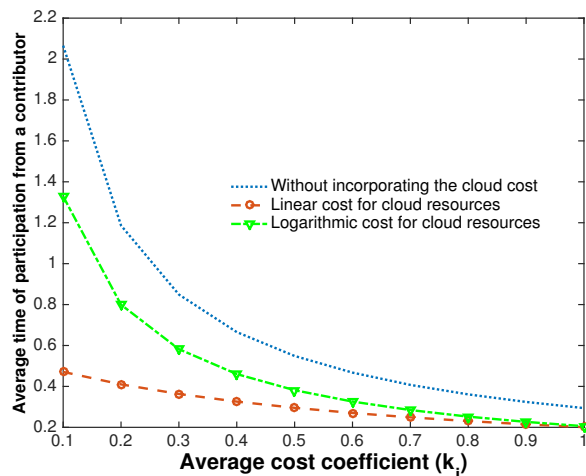


Fig. 7. Average service time of the crowdsensors w.r.t. average cost coefficient k_j

VI. CONCLUSION

In this paper, we have designed an optimal incentive mechanism in a cloud-enabled multimedia crowdsourcing framework, where the crowdsourcer optimizes its utility over the reward to the contributors, and the contributors optimize their utilities over their duration of service for multimedia

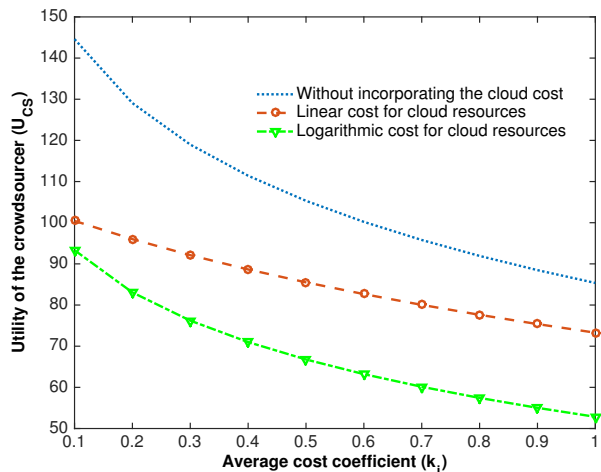


Fig. 8. Utility of the crowdsourcer w.r.t. average cost coefficient k_j

generation. We have incorporated the cost of using the cloud resources for multimedia crowdsourcing applications, into our model. We have derived a close form expression for the optimal solution of the competition among the smartphones, for a given reward from the crowdsourcer. Consequently, we have derived the conditions for the existence of the optimal reward, and have proven its uniqueness.

We believe that this work opens the door to some interesting extensions. The analysis incorporating multiple crowdsourcers is a potential direction. As the problem includes crowdsourcers' heterogeneity in addition to the contributors', the problem would be interesting and timely. In our work we have focused on cloud enabled multimedia crowdsourcing applications in general. Characterizing the specific features of a particular application is another possible extension of this work. Specific application tailored analysis may yield particular and interesting insights.

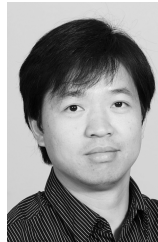
REFERENCES

- [1] Jeff Howe. The rise of crowdsourcing. *Wired Magazine*, 14(6):1–4, 2006.
- [2] N. Nandan, A. Pursche, and Xing Zhe. Challenges in crowdsourcing real-time information for public transportation. In *Mobile Data Management (MDM), 2014 IEEE 15th International Conference on*, volume 2, pages 67–72, July 2014.
- [3] D. Wu, Y. Zhang, L. Bao, and A.C. Regan. Location-based crowdsourcing for vehicular communication in hybrid networks. *Intelligent Transportation Systems, IEEE Transactions on*, 14(2):837–846, June 2013.
- [4] Anmol Madan, Manuel Cebrian, Sai Moturu, Katayoun Farrahi, and Alex Sandy Pentland. Sensing the health state of a community. *IEEE Pervasive Computing*, 11(4):36–45, 2012.
- [5] Xiaonan Guo, E.C.L. Chan, Ce Liu, Kaishun Wu, Siyuan Liu, and L.M. Ni. Shopprofiler: Profiling shops with crowdsourcing data. In *INFOCOM, 2014 Proceedings IEEE*, pages 1240–1248, April 2014.
- [6] G. Markowsky. Crowdsourcing, big data and homeland security. In *Technologies for Homeland Security (HST), 2013 IEEE International Conference on*, pages 772–778, Nov 2013.
- [7] Tinghuai Ma, Jinjuan Zhou, Meili Tang, Yuan Tian, Abdullah Al-Dhelaan, Mznah Al-Rodhaan, and Sungyoung Lee. Social network and tag sources based augmenting collaborative recommender system. *Information and Systems, IEICE Transactions on*, E98-D(4):902–910, April 2015.
- [8] Chao Song, Jie Wu, Ming Liu, Haigang Gong, and Bojun Gou. Resen: Sensing and evaluating the riding experience based on crowdsourcing by smart phones. In *Mobile Ad-hoc and Sensor Networks (MSN), 2012 Eighth International Conference on*, pages 147–152, Dec 2012.

- [9] Chenshu Wu, Zheng Yang, and Yunhao Liu. Smartphones based crowdsourcing for indoor localization. *Mobile Computing, IEEE Transactions on*, 14(2):444–457, Feb 2015.
- [10] A. Faggiani, E. Gregori, L. Lenzini, V. Luconi, and A. Vecchio. Smartphone-based crowdsourcing for network monitoring: Opportunities, challenges, and a case study. *Communications Magazine, IEEE*, 52(1):106–113, January 2014.
- [11] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, and D. Zenalipour-Yazti. Crowdsourcing with smartphones. *Internet Computing, IEEE*, 16(5):36–44, Sept 2012.
- [12] Bin Guo, Zhiwen Yu, Daqing Zhang, and Xingshe Zhou. From participatory sensing to mobile crowd sensing. *CoRR*, abs/1401.3090, 2014.
- [13] Bin Guo, Zhu Wang, Zhiwen Yu, Daqing Zhang, and Xingshe Zhou. Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm. *ACM Comput. Surv.*, 48(1), August 2015.
- [14] D. Estrin, K. M. Chandy, R. M. Young, L. Smarr, A. Odlyzko, D. Clark, V. Reding, T. Ishida, S. Sharma, V. G. Cerf, U. Hitzle, L. A. Barroso, G. Mulligan, A. Hooke, and C. Elliott. Participatory sensing: applications and architecture [internet predictions]. *IEEE Internet Computing*, 14(1):12–42, Jan 2010.
- [15] YouTube. <https://www.youtube.com/>.
- [16] Baochun Li, Zhi Wang, Jiangchuan Liu, and Wenwu Zhu. Two decades of internet video streaming: A retrospective view. *ACM Trans. Multimedia Comput. Commun. Appl.*, 9(1s):33:1–33:20, October 2013.
- [17] RapidShare. <https://en.wikipedia.org/wiki/RapidShare>.
- [18] facebook. <https://www.facebook.com/>.
- [19] Netflix. <https://www.netflix.com/no-en/>.
- [20] Stream2Watch. <http://www.stream2watch.me/>.
- [21] sportLEMON. sportLEMON.tv/c.html.
- [22] Twitch TV. <http://www.twitch.tv/>.
- [23] P. Mell and T. Grance. The NIST Definition of Cloud Computing. <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>, Sept 2011.
- [24] Yonggang Wen, Weiwen Zhang, and Haiyun Luo. Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones. In *INFOCOM, 2012 Proceedings IEEE*, pages 2716–2720, March 2012.
- [25] Zhangjie Fu, Xingming Sun, Qi Liu, Lu Zhou, and Jiangang Shu. Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing. *Communications, IEICE Transactions on*, E98-B(1):190–200, January 2015.
- [26] Fei Chen, Cong Zhang, Feng Wang, and Jiangchuan Liu. Crowdsourced live streaming over the cloud. *CoRR*, abs/1502.06314, 2015.
- [27] Dejun Yang, Guoliang Xue, Xi Fang, and Jian Tang. Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Mobicom '12*, pages 173–184, New York, NY, USA, 2012. ACM.
- [28] Yao Sun, Yanmin Zhu, Zhenni Feng, and Jiadi Yu. Sensing processes participation game of smartphones in participatory sensing systems. In *Sensing, Communication, and Networking (SECON), 2014 Eleventh Annual IEEE International Conference on*, pages 239–247, June 2014.
- [29] R.K. Ganti, Fan Ye, and Hui Lei. Mobile crowdsensing: current state and future challenges. *Communications Magazine, IEEE*, 49(11):32–39, November 2011.
- [30] Dong Zhao, Xiang-Yang Li, and Huadong Ma. How to crowdsourcing tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint. In *INFOCOM, 2014 Proceedings IEEE*, pages 1213–1221, April 2014.
- [31] Zhenni Feng, Yanmin Zhu, Qian Zhang, Hongzi Zhu, Jiadi Yu, Jian Cao, and L.M. Ni. Towards truthful mechanisms for mobile crowdsourcing with dynamic smartphones. In *Distributed Computing Systems (ICDCS), 2014 IEEE 34th International Conference on*, pages 11–20, June 2014.
- [32] Huadong Ma, Dong Zhao, and Peiyan Yuan. Opportunities in mobile crowd sensing. *Communications Magazine, IEEE*, 52(8):29–35, Aug 2014.
- [33] Ju Ren, Yaoxue Zhang, Kuan Zhang, and Xuemin (Sherman) Shen. Sacrm: Social aware crowdsourcing with reputation management in mobile sensing. *Computer Communications*, 65:55 – 65, 2015. Mobile Ubiquitous Sensing from Social Network Viewpoint.
- [34] G. Markowsky. Crowdsourcing, big data and homeland security. In *Technologies for Homeland Security (HST), 2013 IEEE International Conference on*, pages 772–778, Nov 2013.
- [35] Wenwu Zhu, Chong Luo, Jianfeng Wang, and Shipeng Li. Multimedia cloud computing. *Signal Processing Magazine, IEEE*, 28(3):59–69, May 2011.
- [36] Yonggang Wen, Xiaoqing Zhu, J.J.P.C. Rodrigues, and Chang Wen Chen. Cloud mobile media: Reflections and outlook. *Multimedia, IEEE Transactions on*, 16(4):885–902, June 2014.
- [37] P. Simoens, Y. Xiao, Pillai P., Chen Z., Ha K., and Satyanarayanan M. Scalable crowd-sourcing of video from mobile devices. In *MobiSys, 2013 Proceedings*, pages 139–152, June 2013.
- [38] Jian Li, Xiaolong Li, Bin Yang, and Xingming Sun. Segmentation-based image copy-move forgery detection scheme. *Information Forensics and Security, IEEE Transactions on*, 10(3):507–518, March 2015.
- [39] Zhaoqing Pan, Yun Zhang, and Sam Kwong. Efficient motion and disparity estimation optimization for low complexity multiview video coding. *Broadcasting, IEEE Transactions on*, 61(2):166–176, June 2015.
- [40] C. Keimel, J. Habigt, C. Horsch, and K. Diepold. Video quality evaluation in the cloud. In *2012 19th International Packet Video Workshop (PV)*, pages 155–160, May 2012.
- [41] White Paper: Understanding and Reducing Latency in Video Compression Systems. <http://www.cast-inc.com/blog/white-paper-understanding-and-reducing-latency-in-video-compression-systems>, Oct 2013.
- [42] A. Bashandy, E. Chong, and A. Ghafoor. Network modeling and jitter control for multimedia communication over broadband network. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 559–566 vol.2, Mar 1999.
- [43] Google. <https://cloud.google.com/pricing/>.



Sabita Maharjan (S'09-M'13) is currently a Post-doctoral Fellow in Simula Research Laboratory, Norway. She received M.Eng. degree in wireless communications from Antenna and Propagation Lab, Tokyo Institute of Technology, Japan, in 2008 and Ph.D. degree in Networks and Distributed Systems from University of Oslo and Simula Research Laboratory, Norway, in 2013. Her current research interests include resource optimization in wireless networks, network security, game theory, smart grid communications and Internet of things.



Yan Zhang (SM'10) is a full-time Professor at the Department of Informatics, University of Oslo, Norway. He is also a Chief Research Scientist at Simula Research Laboratory, Norway. He received a PhD degree in School of Electrical & Electronics Engineering, Nanyang Technological University, Singapore. He is IEEE Access Associate Editor. He also serves as the guest editor for IEEE Communications Magazine, IEEE Wireless Communications Magazine, IEEE Network Magazine, IEEE Transactions on Smart Grid, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Industrial Informatics, IEEE Systems Journal, IEEE Access, and IEEE Internet of Things journal. He serves as chair positions in a number of conferences, including IEEE PIMRC 2016, IEEE CloudCom 2016, IEEE ICC 2016, IEEE CCNC 2016, WCSP 2016, WICON 2016, IEEE SmartGridComm 2015, and IEEE CloudCom 2015. He serves as TPC member for numerous international conferences including IEEE INFOCOM, IEEE ICC, IEEE GLOBECOM, and IEEE WCNC. His current research interests include: next-generation wireless networks leading to 5G, reliable and secure cyber-physical systems (e.g., smart grid, healthcare, and transport), and Internet-of-Things. He is IEEE VTS (Vehicular Technology Society) Distinguished Lecturer. He is also a senior member of IEEE, IEEE ComSoc, IEEE PES, and IEEE VT society. He is a Fellow of IET.



Stein Gjessing (M'90) Stein Gjessing received his dr. philos. degree in 1985, and is currently a professor of Computer Science in Department of Informatics, University of Oslo, and an adjunct researcher at Simula Research Laboratory. Gjessings original work was in the field object oriented concurrent programming. He has worked with computer interconnects (Scalable Coherent Interface (IEEE Std. 1596) and LAN/MANs (Resilient Packet Ring (IEEE Std. 802.17)). His main research interests are currently network and transport protocols, network resilience, cognitive radio networks, vehicular networks and the smart grid.