Cost-Benefit Tradeoffs of Content Sharing in Personal Cloud Storage

Glauber Gonçalves*, Alex Borges Vieira†, Idilio Drago‡
Ana Paula Couto da Silva*, Jussara M. Almeida*

*Universidade Federal de Minas Gerais, Brazil
†Universidade Federal de Juiz de Fora, Brazil
‡Politecnico di Torino, Italy
ggoncalves@dcc.ufmg.br; alex.borges@ufjf.edu.br; idilio.drago@polito.it
ana.coutosilva@dcc.ufmg.br; jussara@dcc.ufmg.br

***Abstract*—Personal Cloud Storage (PCS) is a very popular Internet service. It allows users to backup data to the cloud as well as to perform collaborative work while sharing content. Notably, content sharing is a key feature for PCS users. It however comes with extra costs for service providers, as shared files must be synchronized to multiple user devices, generating more downloads from cloud servers. Despite the increasing interest in this type of service, a thorough investigation on the costs and benefits of PCS for service providers and end users has not been conducted yet. To that end, we propose a model to analyze cost-benefit tradeoffs for both parties. We develop utility functions that capture, in an abstract level, the satisfaction of the service provider and users in various scenarios. Then, we apply our model to evaluate alternative policies for content sharing in PCS. We consider two alternative policies for the current PCS sharing architecture, which count on user collaboration to reduce providers' costs. Our results show that such policies are advantageous for providers and users, leading to 39% utility improvements for both parties, while requiring low commitment of resources from participating users.**[1]

I. Introduction

Personal Cloud Storage (PCS) [5], [11], [8] is a popular Internet service. Well-known applications, such as Dropbox, Google Drive and Microsoft OneDrive, reached close to 36% of broadband users in certain regions in 2015 [1]. Such services offer users the benefits of backing up content with great simplicity to the cloud. Additionally, they allow content sharing among multiple user devices with support of the cloud, thus enabling the user to synchronize files among her devices as well as to perform collaborative work with other users in almost real-time.

Content sharing has become a valuable feature for users of PCS. Indeed, recent studies show evidence of massive adoption of content sharing by PCS users [7], [9], [18]. By promoting user interactions driven by social ties or facilitating content organization across multiple devices, content sharing may contribute to increase user satisfaction with the service. However, this functionality comes with extra costs for the service provider. The synchronization of files to multiple devices associated with the sharing generates extra downloads from the cloud servers to such devices, which ultimately incur in more resources (notably bandwidth) required from the provider infrastructure. For example, the fraction of PCS traffic related to downloads is often quite large – e.g., 68% of Dropbox traffic in some networks as we will show later.

This issue raises questions about the costs and benefits of content sharing in PCS for both end users and the service provider. Most providers adopt pricing models that allow free storage usage with limited space, charging for extra space. The idea behind such model is that the service can attract more (paying and free) users while remaining profitable, as the payments for extra space compensate the overall costs. However, such model clearly pressures the provider to reduce costs in order to maintain profitability, since the fraction of users who pay for the service is often small. On the other hand, the policies adopted to reduce costs must not hurt user satisfaction, at the penalty of reducing service attractiveness and losing the paying users.

Indeed, although some providers have survived even with very small fraction of paying users (e.g., 4% in Dropbox [19]), others have left the market (e.g., UbuntuOne, Wualla), possibly due to high operational costs [13], [22]. Content sharing may exacerbate the problem. Ultimately, these dynamics generate complex tradeoffs between costs and benefits for both users and providers. Identifying strategies that solve such tradeoffs by balancing the satisfaction of both parties is of utmost importance for providers to remain competitive in the market.

Most prior studies that have investigated these tradeoffs analyzed costs and benefits from a single point of view, either focusing on end users [17], [21] or service providers [24], [12]. A prior effort that jointly analyzed user and provider perspectives has focused on general storage services and does not take into account the costs associated with shared content [15], which is an important concern in PCS. Costs and benefits of content sharing have only been discussed in our prior work [7], though still with focus only on the provider side. Thus, to our knowledge, no prior work tackled the cost-benefit tradeoffs in PCS, considering content sharing, while meeting the interests of both users and service providers.

[1] A preliminary version of this work has appeared in [6].

2375-0227/17 $31.00 © 2017 IEEE
DOI 10.1109/MASCOTS.2017.32

We here investigate cost-benefit tradeoffs of content sharing in PCS for the provider and users jointly. We start from the following broad question: *How to model costs and benefits of PCS so to help providers in assessing the effectiveness of alternative policies that aim at increasing both profitability and user satisfaction?* Various characteristics of PCS should be taken into account when tackling this question. Some examples are the majority of free users in the service, the attractiveness of content sharing for users (and the corresponding costs for provider), and the urge to keep users satisfied in such a competitive market. The inter-dependencies among these characteristics make the investigation quite challenging.

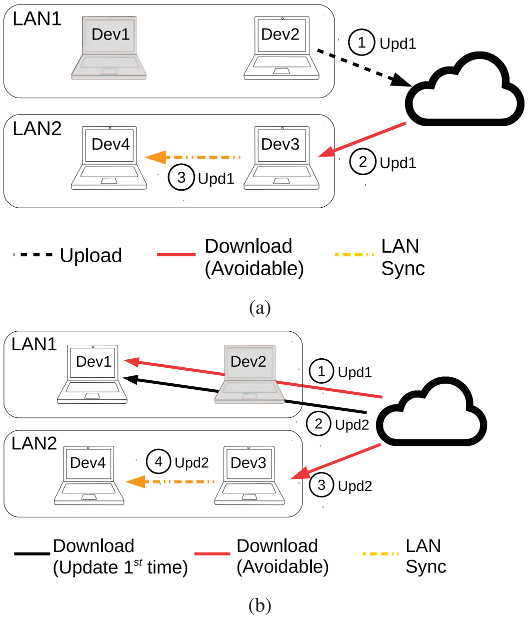
We make two contributions:

- We propose a general model for the costs and benefits of PCS considering both users and providers. To that end, we propose *utility functions* that represent the benefits minus the costs of the service for each party. The greater the utilities are, the more satisfied providers and users become. To keep our scope limited, we focus mostly on objective aspects of PCS (e.g., storage and bandwidth costs), ignoring more subjective aspects (e.g., social impact of collaborative work). As such, our proposal is appealing for capturing in a simple (but representative) model key components of PCS.
- We investigate two alternative policies for the current PCS sharing architecture, which count on user collaboration to reduce provider costs. We evaluate the effectiveness of both policies by applying our proposed model to assess user and provider utilities in various scenarios.

Our analyses are performed using traces of Dropbox usage collected in different networks, where around 23 TB of Dropbox traffic and 27 428 unique user devices have been observed. We have chosen Dropbox as case study as it is currently one of the leaders in the market [1], [5], [14], with more than 500 million users and 3 billion sharing connections.[2] Nevertheless, the proposed policies are applicable to other services as well.

Our results show that the investigated policies are advantageous for providers and users, leading to utility improvements of up to 39% for both parties with respect to current settings. Improvements come from offloading the provider infrastructure by counting on users to help transferring shared content via a Peer-to-Peer (P2P) architecture. This strategy becomes attractive to users as the provider gives bonus to those who collaborate. Best utility improvements are achieved with a limited number of collaborators (around 30% of the eligible devices). Gains are more relevant in scenarios where users share lots of contents, but those collaborating still need to contribute with only a small amount of resources – e.g., no more than 23% of the Internet bandwidth they are willing to offer to the provider.

Next, Section II presents the background on PCS services and content sharing in Dropbox. In Section III, we model costs and benefits of general PCS (provider and user utilities). We describe alternative content sharing policies and respective model variations in Section IV. In Section V, we apply the model to evaluate the policies. We discuss related work in Section VI, and finally conclude the paper in Section VII.



(a)

(b)

Figure 1. Examples of downloads in PCS due to content sharing. An update in a shared folder is generated (a) inside the given network (Device 1 is offline) and (b) outside the network (Device 2 is offline, Device 1 is back online and now has to retrieve two updates from the cloud).

II. Content Sharing in PCS

We provide a short background on content sharing mechanisms adopted by many PCS services (Section II-A), followed by measurement results regarding Dropbox (Section II-B), which motivate and drive following analyzes.

A. Sharing Mechanisms

Users of PCS services usually can register multiple devices in the system. Most services require users to have an initial sync-point in their devices – i.e., a local folder from where content is synchronized with the cloud. This folder becomes visible to any other registered device belonging to the same given user. Additionally, users might share content with others, by creating folders or by selecting particular files that are visible to third-parties. Shared content immediately becomes available to all other participating users, which can decide to synchronize the content with their personal devices too.

The protocol to synchronize devices varies from service to service. Popular offers such as Dropbox and Microsoft OneDrive employ a mechanism that triggers synchronization in desktop and laptop clients almost instantaneously. Devices registering the shared resources retrieve the content immediately if on-line, or as soon as they come back on-line, although users can manually pause/stop the automatic content synchronization. It is thus clear that such mechanisms lead to multiple downloads of a single content from cloud servers by the several user devices participating in the sharing.

[2] https://www.dropbox.com/about

Table I
SHARING USING DROPBOX IN FOUR DIFFERENT NETWORKS.

	# users who share some content	**Download traffic (GB)***		
		Volume (% of traffic)	**Shared content (% of downloads)**	**Avoidable (% of downloads)**
Campus-1	3,437 (56%)	2,282 (60%)	1,425 (62%)	411 (18%)
Campus-2	856 (65%)	443 (44%)	253 (57%)	74 (17%)
PoP-1	2,398 (44%)	3,909 (57%)	2,601 (67%)	761 (19%)
PoP-2	828 (46%)	2,558 (68%)	1,801 (70%)	637 (25%)
Total	**7,519 (51%)**	**9,192 (60%)**	**6,080 (66%)**	**1,883 (20%)**

* Volume refers to the sample of users for which we can estimate the download traffic (see [7]).

Table II
SUMMARY OF THE MODEL NOTATION.

Variable	Description
U_s	Service provider utility
U_i	User i utility
$\mathcal{R}$	Provider's revenue including users' payments and other sources
V_i	Valuation of user i per byte – e.g., bytes stored in the cloud, or bytes offloaded from the provider by user i
α	Price of one byte of storage for the provider
β	Price of one byte transferred from/to the cloud for the provider
P_i	Price the user i pays for service
X_i	Storage capacity (in bytes) available to user i in the cloud
$\mathcal{S}$	Total bytes stored in the cloud by all users
$\mathcal{T}$	Total bytes transferred from/to the cloud by all users
$\mathcal{O}$ ($\mathcal{O}_i$)	Total volume (in bytes) offloaded from provider (by user i)
κ	Units of bonus offered by the provider to users per offloaded byte
$\mathcal{C}_i$	Penalty factor imposed on user i per offloaded byte
$\mathcal{B}_i$	Maximum upload capacity (in bytes) user i is willing to offer for offloading the provider
$\mathcal{D}_i$	Maximum storage capacity (in bytes) user i is willing to offer for offloading the provider

Some services have protocols in place to reduce download costs by performing device-to-device synchronization. Dropbox's LAN Sync protocol is the most prominent example [4]. However, the protocol operates in a rather limited scope. Figure 1 illustrates how content download happens in Dropbox. Dropbox's LAN Sync only synchronizes devices located in the same Local Area Network (LAN). In the example, four different devices are connected to two LANs in a single network (e.g., in a campus). The devices share content and are kept synchronized by retrieving updates either directly from the cloud (black or red arrows) or from local peers using the LAN Sync protocol (orange arrows). As the LAN Sync is a P2P protocol, devices have to be online *simultaneously* to profit from it.

As the figure shows, some updates (solid red arrows) need to be retrieved from the cloud, even if the same updates have already been observed in the network. Update 1, performed by Device 2 while Device 1 is offline (Figure 1-a), is retrieved from the cloud by Device 1 when it comes back on-line, because Device 2 has become unavailable in the meanwhile (Figure 1-b). Note that these downloads of content already seen in the network (called *avoidable downloads* in [7]) may also occur when multiple devices need to retrieve updates generated elsewhere (e.g., Update 2 in Figure 1-b).

Therefore, avoidable downloads occur even if devices are close to each other, e.g., multiple users connected to distinct LANs in the same network, or clients of different Internet Service Providers (ISP) living in the same neighborhood. As we will argue next, the synchronization of shared content is responsible for a major portion of the PCS traffic.

B. Sharing in Dropbox

We use the real traces of Dropbox traffic first appearing in our previous work [7], [8].[3] Data about Dropbox usage have been collected during 12 months at four vantage points, including two university campuses and two ISP networks. The datasets expose a list of (anonymized) shared folder, device and user IDs, along with traffic statistics. Thus, it allows us to study content sharing in Dropbox, while offering no hints about users' identities or the content stored in the service.

We have developed in the previous work a methodology to (i) estimate the amount of content transferred by each user device and (ii) determine whether the content is related to *shared folders*, i.e., the content is shared among several devices of the same user or multiple users. In short, Dropbox clients used to announce in the network the version number for each shared folder present in users' devices. By observing the traffic towards Dropbox servers, we could directly know when shared folders were changed, and estimate the volume of each update. Moreover, by tracking user and device IDs, we could calculate statistics about the number of users participating in each sharing, and determine if a particular update in a shared folder was downloaded by different devices in the network.

Table I summarizes the observed volume of content sharing via Dropbox. A relevant percentage of users (44–65%) is associated with at least one shared folder. Generally, download volumes are very high, and sometimes higher than upload volumes. Since downloads from the cloud are necessarily a consequence of sharing among devices, we can safely conjecture that costs of such functionality are significant for providers. Note that 44–68% of Dropbox traffic is download from the cloud.

More than that, we confirm that up to 70% of these downloads are related to shared folders in the same network – i.e., we are able to link them to a folder seen in at least two devices. By monitoring the version number of shared folders in different devices, we conclude that avoidable downloads (i.e., a single update going to more than one device) are very common. The rightmost column in Table I reports that up to 25% of the Dropbox incoming traffic falls into this category.

III. GENERAL COST-BENEFIT MODEL

We now introduce our model for costs and benefits of PCS, covering both the service provider and users. We aim at developing a simple but reasonably representative model that captures, to a good extent, the main components of a PCS service that drive satisfaction of both parties. Thus, we focus on aspects related to resource consumption, notably storage

[3] Datasets are available at http://locus.dcc.ufmg.br/datasets/pcssharing.html.

and bandwidth. Other aspects, such as the speed of upload or download and the social value of collaborative work, are left out for limiting the present scope.

Our model is based on utility functions that capture, in an abstract level, the provider profit and the user surplus (i.e., satisfaction) with the service. As in other studies [25], [20], [15], these functions express the difference between the benefits and the costs of the service for each party. Table II lists the main notation used in the models presented here and in Section IV. We note that costs and benefits (and corresponding utilities) may vary over time. We then assume all model variables are expressed for the time window w (e.g., a week or a month). Resources consumed before/after w as well any feedback effect due to decisions made during w (e.g., user response to benefits received) are left out of the present analysis. For simplicity, we omit the time window from the notation, and present the model for any arbitrary time window.

We consider a monopolistic service provider, which charges a fixed monthly or annual price for a certain storage capacity per user (paying users). This provider also offers free storage with a small capacity to any user who registers in the service (free user). Under this pricing model, the utility of the service provider, U_s, can be defined as:

$$U_s = \mathcal{R} - (\alpha * \mathcal{S} + \beta * \mathcal{T}), \quad (1)$$

where the provider's benefit is represented by the revenue ($\mathcal{R}$) that comes from paying users and secondary sources. For example, it has been estimated that around 4% of the Dropbox users pay for the service [19], and the company has been supported by funding groups [3].[4] The cost of the provider is estimated in terms of total amount of data stored in the infrastructure ($\mathcal{S}$) and the total amount of data transferred to/from the cloud ($\mathcal{T}$). Both measures are given in bytes, and we define the parameters α and β to represent, respectively, the price per byte of storage per w and the price per byte transferred in the network (i.e., upload/download).

The utility of user i, U_i, is instead given by:

$$U_i = V_i * X_i - P_i, \quad (2)$$

where the user's benefit is represented by her cloud storage capacity (X_i), and a utility level (V_i) that corresponds to the user's valuation for each byte she can store in the cloud. The user's cost is given by the price she pays for the service (P_i).[5] We assume the user valuation can vary according to P_i. For example, paying users ($P_i > 0$) may have a valuation greater than free users.

It is worth noting the conflicting interests represented by these equations. From the provider's point of view, in order to increase utility, it has to (i) grow the revenues by attracting more users, which can increase the service popularity (possibly leading to new investments) or even the fraction of paying users, or (ii) reduce the aforementioned costs. The latter is particularly important as more users lead to more resources required from the infrastructure (larger $\mathcal{S}$ and $\mathcal{T}$). From the user's point of view, instead, the utility can be increased by either uploading more data to the cloud or by looking for lower service prices. Free users ($P_i = 0$), who usually are the majority in PCS services, cannot increase utility beyond the small free storage capacity offered by provider. The challenging task is to reach the best tradeoff between users' and the provider's utilities.

One example of such tradeoff arises when content sharing is taken into account. Offering this functionality is a way of making the service more attractive to users. However, it leads to additional transfer costs (i.e., downloads from the cloud). Defining strategies to reduce such costs is then of utmost importance, as further discussed in the next section.

IV. New Content Sharing Policies

We now build upon the cost-benefit model presented in the previous section and propose content sharing policies that can be advantageous for both the provider and users. The goal of these policies is to enable device synchronization *without the need to retrieve content from the cloud*.[6] We start by presenting our general approach to design such policies and how our general model can be used to estimate the improvements achieved by any such policy (Section IV-A). We then introduce the design principles of two specific policies (Sections IV-B and IV-C). Finally we build upon our general model to derive the impact that each policy causes on provider's and users' utilities (Section IV-D).

A. Estimating Utility Improvements

The provider can apply different strategies in order to jointly increase its utility and users' utility. We evaluate alternative service policies that trigger users to contribute with part of their idle resources (e.g., upstream bandwidth and/or storage), receiving as a compensation a bonus proportional to the costs offloaded from the provider. In this way, the provider can improve its utility by reducing its operational costs, whereas users' utility improvements come from earned bonus.

Specifically, we consider the user bonus as a free extra storage space in the cloud, which the provider attaches to the user account as she offloads the service traffic. Offering bonuses to users has already been explored by some PCS providers. For example, Dropbox offers free storage in campaigns for incentivizing users to invite friends to the service.[7]

Given a policy that offers the aforementioned bonuses to users, the new provider's utility function is defined as:

$$U_s^{new} = U_s + \mathcal{P}_s, \quad (3)$$

where utility gain $\mathcal{P}_s$ added to the previous provider utility U_s comes from the reduction of the provider's cost achieved with the policy. $\mathcal{P}_s$ includes costs related to resources that are

[4] We assume such funding as a secondary revenue source, which allows the provider to cover free users' operating costs, thus increasing its user base.

[5] Note that we ignore the costs of sending the personal content to the cloud, which is the onus applicable to every user registering to the service.

[6] In the case of Dropbox, the policies should enable synchronization in scenarios where `LAN Sync` is not effective, e.g., users in different LANs.

[7] A user who invites someone to the service receives 500 MB of extra space in the cloud when the invited person installs the Dropbox client.

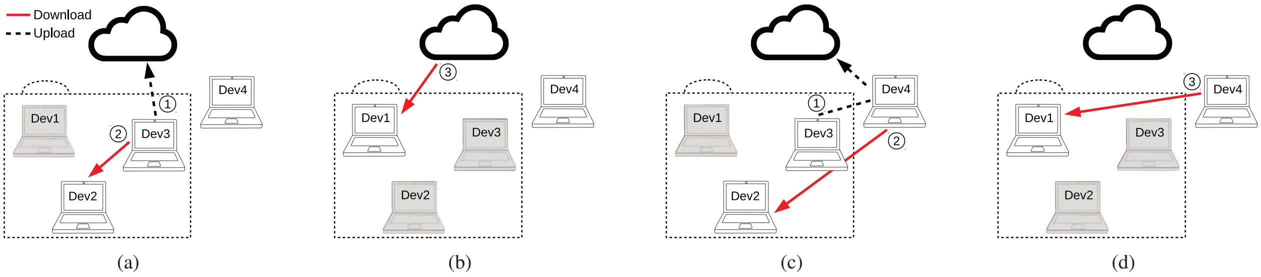


Figure 2. Common cases of downloads in PCS with (a and b) Contact Sharing Policy and (c and d) Anonymous Sharing Policy: Dev 1, 2 and 3 are associated with the same shared folder, (a) Dev 3 sends new updates to the cloud and serves Dev 2 (Dev 1 is offline), (b) Dev 1 is back online and retrieves new updates from the cloud (Dev 2 and 3 are offline), (c) Dev 3 sends new updates to the cloud through Dev 4 (always online but not associated with the folder) which serves Dev 2 (Dev 1 is offline), (b) Dev 1 is back online and retrieves new updates from the Dev 4 (Dev 2 and 3 are offline).

offloaded by users as well as costs related to the new bonuses offered to the participating users.

The new user utility, in turn, is given by:

$$U_i^{new} = U_i + \mathcal{P}_i, \tag{4}$$

where $\mathcal{P}_i$ represents the net benefit earned by user i from her participation in the policy, which includes the bonuses received as well as the new costs related to local resources (storage and network) used for offloading the provider.

The exact definitions of $\mathcal{P}_s$ and $\mathcal{P}_i$ depends on the specific policy adopted. Given these definitions, we are able to estimate the impact of a particular policy in terms of changes in the utilities of the provider and of user i, I_s and I_i, respectively. Such impacts are computed as the relative difference between the new and previous utilities:

$$I_s = \frac{U_s^{new} - U_s}{U_s} = \frac{\mathcal{P}_s}{U_s}; \qquad I_i = \frac{U_i^{new} - U_i}{U_i} = \frac{\mathcal{P}_i}{U_i}. \tag{5}$$

Note that I_s and I_i can become negative if the policies lead to negative impact on the utilities.

Having discussed how we estimate the impact of new content sharing policies on both utilities, we now turn to two specific policies, referred to as Contact Sharing Policy (CSP) and Anonymous Sharing Policy (ASP), presenting their main design principles.

B. Contact Sharing Policy (CSP)

CSP allows users associated with a shared folder, here referred to as *contacts*, to serve each other the content generated in the folder, thus offloading the provider of such data transfers. To accomplish this policy, the provider first establishes a set of eligible user devices for each given folder. Only eligible devices are allowed to serve the folder's content to others. In general, all devices that share the given folder are eligible to serve it. The provider then invites eligible users to participate in the policy, expecting that a percentage of them will accept the invitation and join the policy. Users that accept the offer inform the provider the maximum upload capacity in bytes they are willing to contribute in the time window ($\mathcal{B}_i$).

Figures 2 (a and b) illustrate the main synchronization steps with the CSP policy depending on whether participating devices are online or not. A device always sends content updates created locally[8] to the cloud (step 1). As in most PCS synchronization protocols, control servers keep track of all updates each device owns and notify other online devices about new updates. Indeed, most PCS services employ control servers for keeping track of online devices and of the updates in content metadata. For example, in Dropbox, clients and control servers exchange periodic "keep alive" messages, and the client informs servers of new updates whenever available.

In order to implement CSP, control servers should include, in each update notification sent to a device d, the identifier of one other device that is currently online and can serve the update, if one is available. The control server selects this *source* device s among all online devices that currently have the same update. The selection policy may be random or any other approach, aiming at keeping the load balanced across participating devices. As soon a device d receives the update notification, it tries to establish a connection with the indicated device s in order to download the update. If the download succeeds (step 2), d informs the control server the identifier of s, so that the server can update the bonus to be given the user who owns s accordingly. If the download fails, or no online device to serve the update was available, d retrieves the content directly from the cloud (step 3). After retrieving the content, either from another device s (contact) or the cloud, device d becomes able to serve it.

C. Anonymous Sharing Policy (ASP)

ASP also allows users to serve each other content. However, unlike CSP, the user devices eligible to serve content from a folder do not need to be associated with (i.e., share) the folder, or any other shared folder. In this case, the provider may invite user devices that are often online to participate in this policy, thus increasing the chances of offloading downloads. As for the previous policy, users accepting the offer inform the maximum upload capacity they are willing to contribute ($\mathcal{B}_i$). Moreover, since participants do not own the content they will

[8] Updates represent modifications in shared folders: new files, metadata and the commands that manipulate files, e.g., to delete files or create sub-folders.

offload from the provider, they must also inform the provider the maximum local storage they are willing to contribute ($\mathcal{D}_i$).

Figure 2 (c and d) illustrates the main synchronization steps with ASP into play in the same cases shown in Figure 2 (a and b). The policy works as follows. The devices associated with shared folders receive from the control server a list with a random subset of all participating devices. The list contains the devices that will be responsible for storing content updates. We refer to them as *anonymous* devices. The list of anonymous devices may change periodically so as to reflect the availability and promote load balancing of the participating devices.

A device s sends an update it generates to one of the anonymous devices, which is responsible for forwarding it to the cloud (step 1). The update is concluded only after s receives a confirmation from the control server. Otherwise, device s sends the update directly to the cloud. Control servers should keep track of all updates stored at each anonymous device. This does not represent a significant extra cost to the servers, as they already have to keep track of all content owned by the anonymous device. The rest of the policy is very similar to CSP. Control servers notify other devices about the online anonymous devices that can serve content updates by providing the identifier of one such *source* devices in each update notification. As soon as a device d receives a new update notification, it tries to establish a connection with the indicated source device s in order to download the update. If the download succeeds (steps 2), device d informs the control server the anonymous device s that served it. Otherwise, device d retrieves the content from the cloud. Note that, unlike in CSP, when a device comes back online and receives a notification of updates in one of its shared folders, it may be able to retrieve the content from an anonymous device even if none of its contacts are currently online (step 3).

Note also that the proposed design of ASP brings up two issues that are not found in CSP. First, users participating in ASP have an additional disk cost to store other users' data. Second, users are susceptible to privacy violations when forwarding their data to anonymous devices, since such devices may belong to unknown users (unlike in CSP, where data is exchanged only between contacts). The first issue has a direct impact on how user utilities are computed, as we will discuss later. Regarding privacy concerns, we argue that each update must be encrypted before being uploaded to an anonymous device (step 1), as in similar approaches adopted by some PCS services [16]. An investigation of the efficiency of alternative encryption mechanisms in this context is left for future work.

D. Service Cost Reduction and User Bonus

Building upon our model, notably Equations 3 and 4, we now derive the expressions for the cost reduction $\mathcal{P}_s$ experienced by the service provider and the net benefit of a user $\mathcal{P}_i$ for both CSP and ASP. As in Section III, cost reductions and bonuses should be recomputed periodically, for pre-defined time windows w. We present our analysis again focusing on an arbitrary window w.

For both policies, the service provider cost reduction is proportional to the amount of bytes served from client devices, as thus offloaded from the provider's cloud servers during the considered time window. That is:

$$\mathcal{P}_s = (\beta - \alpha * \kappa) * \mathcal{O}, \tag{6}$$

where α and β are the previously defined storage and transfer price units, and κ is the bonus (i.e., number of storage units) the provider offers eligible users per unit of byte offloaded. Note that, while the total amount of bytes offloaded from the server $\mathcal{O}$ causes a reduction on transfer costs (β parameter), it also increases the costs related to storage (α parameter), as extra storage in the cloud servers is offered to participating users (κ bytes per offloaded byte).

We note that the bonus κ expires after a certain period of time (e.g., a time window w)[9]. The provider is able to reach cost reduction by setting $\kappa < \frac{\beta}{\alpha}$. In order to compute the bonus of a given user i, we assume that control servers keep track of the total volume of data served by each device of i participating in the policy. $\mathcal{O}_i$ corresponds to the total number of bytes served by (and thus offloaded from the provider's servers) all devices owned by user i. Similarly $\mathcal{O}$ corresponds to the sum of $\mathcal{O}_i$ for all users with participating devices.

The net benefit of user i for participating in the policy in turn is given by:

$$\mathcal{P}_i = V_i * \mathcal{O}_i * \kappa - V_i * \mathcal{O}_i * \mathcal{C}_i. \tag{7}$$

The utility change for user i is computed from two components. At the one hand, the volume the user has offloaded from the provider $\mathcal{O}_i$ is converted into a bonus using κ, and weighted by the user i's valuation for a byte V_i. On the other hand, from the computed bonus, we discount a penalty related to the user's resources used for offloading the servers. The penalty is also proportional to the bytes offloaded by the user (i.e., $\mathcal{O}_i$), considering a factor $\mathcal{C}_i$, which is the penalty per byte stored/transferred for the provider. Again, user i's valuation for a byte V_i is used as a weight for the penalty.[10]

The above expressions for $\mathcal{P}_s$ and $\mathcal{P}_i$ are the same for both CSP and ASP. The models of the two policies diverge when it comes to the penalty imposed on individual users. User devices participating in CSP only serve content they already share. Thus no extra storage, but rather only transfer capacity, is required from them. In ASP, in turn, user devices store content shared by others. Thus, extra storage is required as well. In both cases, we express the penalty factor imposed on user i, $\mathcal{C}_i$, as the fraction of the available resources that are actually used for offloading the provider.

In other words, given $\mathcal{B}_i$ and $\mathcal{D}_i$ the maximum upload and storage capacity (in total number of bytes) user i is willing to offer for serving others, the penalty factor imposed on a user participating in CSP is defined as $\mathcal{C}_i^{CSP} = \frac{\mathcal{O}_i}{\mathcal{B}_i}$. For ASP,

[9] Different options of bonus usage by users can be adopted, e.g., a single time window w or divided into equal parts by various windows.

[10] Note that we could have used a distinct users' valuation for the penalty, in place of V_i. We opt for a single variable for simplicity.

in turn, the penalty factor is defined as $\mathcal{C}_i^{ASP} = \frac{\mathcal{O}_i}{\mathcal{B}_i} + \frac{\mathcal{O}_i}{\mathcal{D}_i}$. The closer the used resources get to the available capacity, the greater the penalty imposed on users.

V. Evaluation

In this section, we evaluate the efficiency of the two content sharing policies, CSP and ASP, using our model (Equations 5), delimiting specific scenarios where both, provider and users, experience improvements in their utilities. We start by presenting our evaluation methodology (Section V-A) and then discuss our most representative results (Section V-B).

A. Methodology

1) Simulating the Content Sharing Policies: We perform a trace-driven simulation of CSP and ASP policies using Dropbox datasets (Section II-B). Specifically, for each user device in the traces, we track the exact time periods during which the device is connected to the service provider control servers (online) and which periods it is offline. We then select the set of eligible user devices that can participate in the policy. As presented in Section IV, for CSP, the set of eligible devices associated with each shared folder corresponds to all devices that synchronize any content in that particular folder. For ASP, this set corresponds the top $T\%$ devices with longest average online periods. In this case, we assume the users who own those devices may be interested in participating in ASP as they can earn more bonuses due to their availability to offload the provider. We vary user adoption of the policy by varying the fraction of eligible devices who accepts participating in the policy. In our evaluation, we set the number of eligible devices in ASP (T) equal to the total number of eligible devices for CSP, so as to be able to fairly compare both policies.

We then carry out trace-driven simulations of each policy by following each upload/download event observed in the input traces. As described in Section IV, an upload to a shared folder identifies a device that can serve future updates of that content (source device). Thus, whenever there is a download request from a destination device d, we first check whether there is any source device s that owns the update requested in the given download event *and* is currently online. If there is such s, the update is downloaded from source s to destination d. If there are multiple devices that can act as source, we select one randomly. Otherwise the download is served from the cloud, as it happens in the traces currently.

Note that, in our simulations, we can only track potential source devices located within the particular networks covered by the traces we own. As consequence, a source device only will be able to serve updates related to avoidable downloads (see Section II). The generation of updates by devices outside the analyzed networks cannot be tracked, as they are not in the traces. Nevertheless, both CSP and ASP could be applied to offload more downloads from the cloud by tracking potential sources from other (neighboring) network domains. We leave an investigation of the efficiency of both policies in such scenario for the future as it requires traces with a broader view of content sharing in PCS.

2) Reference Setup: Our goal is to investigate scenarios where both the provider and end users are satisfied (utility improvements). To that end, we search for possible scenarios by varying two key parameters, namely, the fraction of eligible user devices that accept to participate in the policy (*policy adoption*), and the units of bonus given to participants per byte offloaded (κ). The remaining parameters are kept fixed, using realistic PCS parameter values, as described next. Each simulation covers a one month time window w, and we compute results for all non-overlapping windows in our traces. In particular, the total number of bytes offloaded by each user ($\mathcal{O}_i$) is dictated by the dynamics of each policy applied on the input traces.

Regarding the provider configuration, we set parameters $\mathcal{S}$ and $\mathcal{T}$ based on our traces to represent realistic cost estimates for a PCS service in the analyzed networks. Specifically, we use the upload and download volumes estimated (as described in [7]) in each trace to define the total number of bytes transferred from/to the cloud ($\mathcal{T}$). Moreover, we add up all folder updates observed in each trace to determine the total number of bytes stored in the cloud ($\mathcal{S}$).

All remaining parameters are set according to values currently adopted by main cloud infrastructure providers and PCS services. For example, we take as reference for α and β the prices for standard storage and bandwidth announced by Amazon S3 at the time of the writing[11]: $\alpha = \$0.03/GB/month$ and $\beta = 3 * \alpha$, i.e., transfer price is three times higher than storage within w. In addition, only traffic from cloud to devices (download) is charged by Amazon.

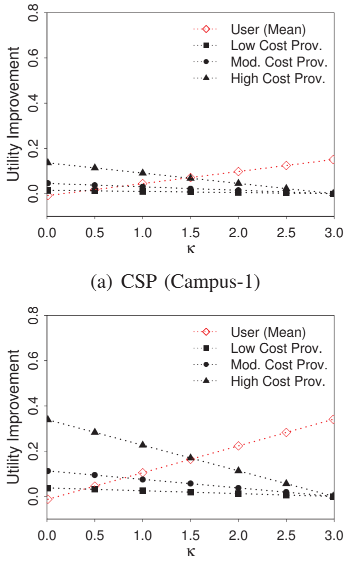
Given the lack of accurate estimates of the sources of revenue for the provider (parameter $\mathcal{R}$), we decide to express it in terms of the provider's total cost (storage and transfers). Specifically, we consider three scenarios for the provider:

- *Low cost*: total cost is 25% of its total revenue;
- *Moderate cost*: total cost is 50% of its total revenue;
- *High cost*: total cost is 75% of its total revenue.

Regarding user configuration, we assume that only free users (P_i=0) may adopt the CSP and ASP policies, as they have more chances of reaching the space limit, and thus become interested in the storage bonuses offered by the policy. Thus, we set the user storage in the cloud based on current free storage capacity adopted by Dropbox ($X_i = 2GB$). Moreover, we assume the same valuation V_i for all users, given the absence of better estimates. Clearly, in real world, the value of each stored byte may vary across different users. However, under a fixed user valuation, we note that our simulation results are not affected by the used parameter value, since we focus on relative utility improvements and consider only free users. Exploring the impact of V_i on the efficiency of policies is left for future work.

Finally, we set the maximum upload capacity $\mathcal{B}_i$ a user is willing to offer in the time window w to 83 GB, which

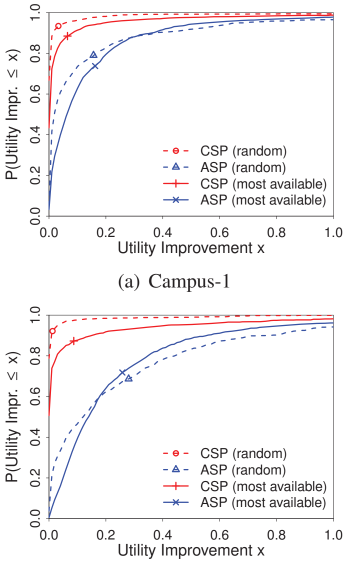
[11] https://aws.amazon.com/s3/pricing



(a) CSP (Campus-1)

(b) ASP (Campus-1)

Figure 3. Utility improvement for the provider and users as functions of κ with 10% policy adoption: best tradeoff occurs at the intersections of the provider and user utility improvement curves.



(a) Campus-1

(b) PoP-2

Figure 4. User utility improvements at target κ for a high cost provider. The average utility improvements are shown as symbols in each curve.

corresponds to the maximum volume a user can transfer in a month under a 256 kpbs connection. Similarly, the maximum storage capacity $\mathcal{D}_i$ is set to 500 GB. Both parameters are conservative, as they correspond to the basic setup offered by main ISPs and PC vendors nowadays.

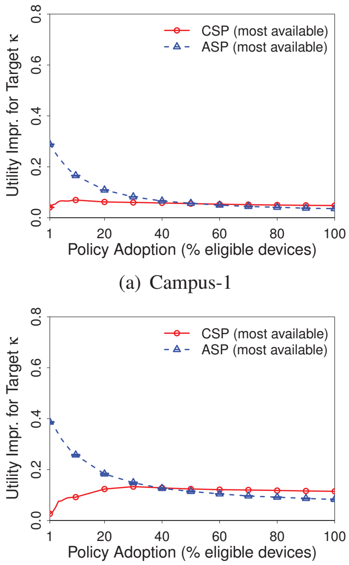
B. *Results*

We analyze the proposed content sharing policies addressing three questions: (Q1) *Is it possible to reach a scenario where both the provider and users obtain utility improvements?* (Q2) *How does the fraction of user devices who adopt the policy impact its efficiency?* and (Q3) *What is the penalty imposed on users that adopt the policy?* We address each question by showing results corresponding to averages computed for all 1 month periods (i.e., w=1 month) in each trace.

1) Utility Tradeoffs: To tackle Q1, we analyze how the units of bonus given to each participant (parameter κ) affect utilities for the provider and users. As one might expect, user utility should increase with κ, whereas the provider utility should decrease with κ. One could then argue that different operation points exist (based on different κ values) where both provider and user experience utility improvements, and selecting the best value is a matter of weighting the satisfaction of both parties. Here we take no side and argue that a "good" operation point for κ is the value for which both provider and (an average) user experience the *same* improvements in their utilities. We refer to this point as the *target* κ. Thus, we search for a κ value where the utility improvements of the provider equal the average utility improvements of a user (average over all users who joined the policy), i.e., $I_s = \bar{I}_i$.

Figures 3-a and 3-b show the utility improvements for both provider and users (on average) as a function of κ for CSP and ASP policies, respectively, on one of our traces (Campus-1). Utility improvements for the three provider scenarios discussed in the previous section are shown in each figure. These results are computed assuming that the 10% eligible devices with the longest average online periods participate in the policy (policy adoption of 10%). Alternatively, adoption could be defined based on a random selection among the eligible devices. However, as we will discuss later, both policies are sensitive to churn in participating devices, i.e., devices often alternating between online and offline states [23]. Thus, we start with a scenario that favors the proposed policies. Results for the other traces are qualitatively similar, and thus are omitted for the sake of brevity.

As shown in the figures, for any scenario – low, moderate or high cost provider – and for any given value of κ, the utility improvements for both provider and user are higher for ASP than for CSP. This is due to the strategy used by ASP to select eligible devices among the most available ones, and not only those that share a folder as in CSP. Thus, even though the approach used to simulate policy adoption is the same, it turns out that the devices participating in ASP tend to be more available in the system, and thus, serve others (offloading the provider) more often. Note that the higher the costs of the provider, the more benefits the provider achieves from using either CSP or ASP, i.e., the higher the utility improvements experienced from applying either policy. For example, considering the high cost provider and ASP, the target κ is approximately 1.5, leading to an improvement of around 17% for both provider and users. In the case of a moderate cost provider, the target κ is 0.8, and the



(a) Campus-1

(b) PoP-2

Figure 5. Utility improvements at target κ for a high cost provider as a function of adopting devices.

improvements of both parties are still relevant, falling around 8%, while in the low cost case, the improvements, although positive (up to 3%), may not be attractive to justify policy deployment.

We delve further into the efficiency of both ASP and CSP from the perspective of individual users by plotting the Cumulative Distribution Function (CDF) of the utility improvements for each individual user (and not only average improvements, as in Figure 3), fixing κ at the target value. Figure 4 shows the distributions for two of our traces, considering a high cost provider. Note that we here present two set of results: one assuming that devices that adopt the policy are selected among the most available ones ("most available" curves) and one assuming that they are selected randomly from the eligible devices ("random" curves). For reference, the average utility improvements are shown as symbols in each curve. Results are very similar for moderate and lower providers as well. Let's focus first on the "most available" adoption scenario. For ASP, around 27% of users experience a utility improvement above average, while for CSP this fraction falls in the 8-13% range. Thus, given the strategy used to select eligible devices, ASP leads to higher improvements for users not only on average but also for individual users. The same is true for the "random" adoption. However, both policies become less effective as some participating devices tend to be less available to serve others.

2) Impact of User Adoption: So far we have assumed a fixed policy adoption at 10%. We now turn to Q2 and analyze the impact this parameter has on policy efficiency. We use the Campus-1 and PoP-2 traces as representative of university and PoP traces, as they have the largest traffic volumes. We also focus on the scenario of a high cost provider, where cost reductions should be more valuable, but the same overall conclusions hold for the other scenarios and traces as well.

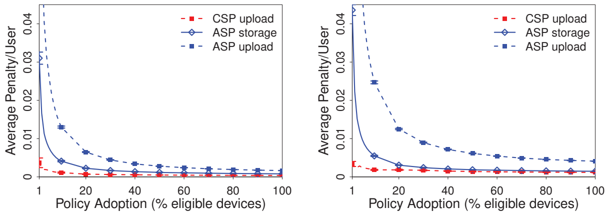
Specifically, we vary the fraction of eligible devices that effectively adopt the policy from 1% to 100% (i.e., all eligible devices), assuming the "most available" case, and measure the utility improvements at the target κ value. For each given fraction, since source devices of policies are non-deterministic, we report average results for 15 independent simulation runs. As shown in Figure 5, improvements are reached for all cases. However, note that the utility improvements experienced under ASP drop sharply as more devices adopt the policy. Given our approach to select adopting devices, as more devices join the policy, it becomes more likely that, at the time of a download event, the device that currently holds the requested update (and thus can serve it) is offline. In other words, the adoption by a larger number of devices with shorter average online period greatly hurts the efficiency of ASP.

For CSP, instead, the utility improvements actually increase with the fraction of adopting devices, dropping slightly or remaining roughly stable after a peak. For example, for the Campus-1 trace, the highest utility improvements – 7% – are obtained with 6% of adopting devices. For PoP-2, 30% adopting devices leads to the highest utility improvements (13%). This happens because when the fraction of adopting devices is too low, multiple devices associated with the same shared folder are rarely simultaneously online to serve each other. Thus, offload opportunities increase as more devices join CSP. Indeed, the total percentage of bytes offloaded from the provider ($\mathcal{O}/\mathcal{T}$ – not shown in the figures) varies from 3% to 15%[12] when the percentage of devices adopting CSP goes from 1% to 100%. However, as more devices participate in CSP, these offloads are distributed across a larger number of source device, and thus the average user utility improvement does not increase accordingly (or may actually slightly drop).

It is also worth noting that the utility improvements are higher in PoP-2 than in Campus-1. This happens because the volumes of updates are typically larger in residential networks (as in PoP-2) than in university networks [8]. The larger the volume of content shared among user devices, the more the provider can reduce its costs with either policy, while the bonuses offered to users as incentive for offloading also increase their satisfaction with the service.

3) Penalty on Participants: Finally, we turn to Q3 and analyze the penalties each policy imposes on users. Figure 6 shows the average value of the penalty factor (parameter C_i) imposed on users participating in CSP and ASP as a function of the fraction of adopting devices, for Campus-1 and PoP-2 traces. For ASP, we separately show the penalty in terms of storage ($\frac{\mathcal{O}_i}{\mathcal{D}_i}$) and upload ($\frac{\mathcal{O}_i}{\mathcal{B}_i}$) resource consumption. As expected, the penalties decrease as more devices join the policy. But, overall, the user resources consumed for offloading the provider tend to be very small, even with low adoption of the policy. This holds for both policies, but especially for CSP. Let's consider the PoP-2 trace, which, as mentioned,

[12] This is 6% to 80% of the avoidable downloads shown in Table I.



(a) Campus-1

(b) PoP-2

Figure 6. Average user penalty vs. policy adoption for a high cost provider.

has the greatest opportunities for offloading. With only 1% of policy adoption, users participating in ASP use, on average, 23% and 4% of their available upload bandwidth and storage, respectively, for offloading the provider. For CSP, the fraction of available upload bandwidth used is only 0.3%.

In sum, our analyses show that, considering as target the same utility improvements for both provider and user, both CSP and ASP lead to significant gains, reaching up to 39%, for both parties. Moreover, the improvements tend to increase with the volume of shared updates, and they come at very low cost for users (in terms of local resource consumption). However, ASP is very sensitive to churn in participating devices. Thus a greater adoption may actually hurt policy efficiency if participating devices are less available for serving each other. Replication of the same update across multiple anonymous devices might minimize this effect, but we leave this investigation for future work.

VI. Related Work

Utility is an important concept in economics and game theory and has already been applied to analyze the relationship between cloud service providers and users [20], [25], [15]. Shen et al. [20] propose utility functions to investigate a win-win scenario in which tenants compete for bandwidth of a provider. Xu and Li [25] study prices of cloud infrastructure resources. Although relevant to our work, these studies do not take into account peculiarities of PCS, such as content sharing. The only prior effort to analyze users and cloud storage providers' utilities [15] (not specifically PCS) focuses on the cost to replicate users' data into multiple disks. Different from our work, authors ignore the role of data transfer costs and content sharing, which are key features to PCS.

Other previous studies focus, separately, on users' or providers' perspectives. Naldi et. al [17] propose a method to compare PCS services based on their pricing policies. Shin et. al [21] investigate users' willingness to pay for PCS, based on consumers' survey and quantitative analyzes. Yeo et. al [26] also explore the users' perspective, proposing a framework that helps users to profit free storage capacity of multiple PCS services. A number of studies explores providers' perspective, mainly proposing methods to maximize the overall profit of providers considering the pricing practices of infrastructure providers [24], [12]. Our work, instead, analyzes cost-benefit tradeoffs for both parties jointly.

The importance of content sharing in PCS has been demonstrated in various studies [5], [11], [1], [9], [18]. Taking the providers' perspective, measurements about Dropbox [5], [1] and UbuntuOne [11] have shown that downloads account for higher traffic than uploads. Taking the users' perspective, Gracia et. al [9] show evidences of massive content sharing adoption by users, and the predominance of content downloaders. A survey conducted in [18] with users of different services identifies content sharing and multi-device synchronization as major reasons for service adoption. These works provide evidences that motivate the study of costs and benefits of PCS, but without addressing the issue as we do in this work.

To our knowledge, only our previous work [7] has addressed the costs and benefits of cloud storage considering content sharing, though still focusing on the provider's point of view. In that work, we have measured the volume of Dropbox updates in real networks and proposed the use of network caches to reduce download traffic. Although efficient, the approach requires investments by the provider to deploy and manage caches. In this work we evaluate a completely opposite approach – i.e., a P2P architecture that involves end-users. We show that this architecture can benefit not only providers, but also end-users by means of the offered bonuses.

The use of P2P in PCS is not new. Wuala was a PCS based on P2P protocols (shut down in 2015). Mager et al. [16] analyzed Wuala, presenting its design and performance. Gracia et al. [10] propose a full P2P-based PCS service, which explores the social ties of users. Chaabouni et al. [2] propose algorithms for bandwidth allocation and P2P swarm management in PCS. None of these studies, however, evaluate the impact of the P2P architecture on costs/benefits of PCS.

VII. Conclusion

This paper investigated cost-benefit tradeoffs of content sharing in PCS for providers and users jointly. We proposed utility functions that represent the benefits minus the costs of the service for each party. Moreover, we investigated two alternative policies for the current PCS sharing architecture, which count on user collaboration to reduce provider costs.

Our analyzes were performed using traces of Dropbox usage collected in different networks. Our results showed that the proposed policies are advantageous for providers and users, leading to utility improvements of up to 39% for both parties with respect to current settings. Gains are more relevant in scenarios where users share lots of contents, and those users collaborating need to contribute with only a small amount of resources. Ultimately, our study calls for service providers to consider further analyzes and the deployment of alternative policies as a competitive advantage to maintain, or even increase, profit and users' satisfaction.

Future directions include analyzing the impact of other variables of our model on utilities. We also intend to apply our model to investigate policies that explore user storage patterns in PCS. This will require further datasets about PCS, beyond content sharing.

References

[1] E. Bocchi, I. Drago, and M. Mellia. Personal Cloud Storage: Usage, Performance and Impact of Terminals. In *Proc. of the IEEE CloudNet*, 2015.

[2] R. Chaabouni, M. Sánchez-Artigas, P. García-López, and L. Pàmies-Juàrez. The power of swarming in personal clouds under bandwidth budget. *Journal of Network and Computer Applications*, 65:48 – 71, 2016.

[3] Crunchbase Inc. Dropbox, 2017. https://www.crunchbase.com/organization/dropbox/funding-rounds.

[4] M. Dee. Inside LAN Sync, 2015. https://blogs.dropbox.com/tech/2015/10/inside-lan-sync.

[5] I. Drago, M. Mellia, M. M. Munafò, A. Sperotto, R. Sadre, and A. Pras. Inside Dropbox: Understanding Personal Cloud Storage Services. In *Proc. of the ACM Internet Measurement Conference*, 2012.

[6] G. Gonçalves, I. Drago, A. Vieira, A. Silva, and J. Almeida. Analysing costs and benefits of content sharing in cloud storage. In *Proc. of the ACM SIGCOMM Workshop on Fostering Latin-American Research in Data Communication Networks (LANCOMM)*, 2016.

[7] G. Gonçalves, I. Drago, A. Vieira, A. Silva, and J. Almeida. The impact of content sharing on cloud storage bandwidth consumption. *IEEE Internet Computing*, 20(4):26–35, 2016.

[8] G. Gonçalves, I. Drago, A. Vieira, A. Silva, J. Almeida, and M. Mellia. Workload models and performance evaluation of cloud storage services. *Elsevier Computer Networks*, 109:183–199, 2016.

[9] R. Gracia-Tinedo, P. García-López, A. Gómez, and A. Illana. Understanding data sharing in private personal clouds. In *Proc. of the IEEE CLOUD*, 2016.

[10] R. Gracia-Tinedo, M. Sánchez-Artigas, A. Ramírez, A. Moreno-Martínez, X. León, and P. García-López. Giving form to social cloud storage through experimentation: Issues and insights. *Future Generation Computer Systems*, 40:1 – 16, 2014.

[11] R. Gracia-Tinedo, Y. Tian, J. Sampé, H. Harkous, J. Lenton, P. García-López, M. Sánchez-Artigas, and M. Vukolic. Dissecting ubuntuone: Autopsy of a global-scale personal cloud back-end. In *Proc. of the ACM Internet Measurement Conference*, 2015.

[12] S. Karunakaran and R. Sundarraj. Bidding strategies for spot instances in cloud computing markets. *IEEE Internet Computing*, 19(3), 2015.

[13] I. Lam. Farewell, wuala! pioneering secure storage shuts down, recommends tresorit, 2015. https://tresorit.com/blog/encrypted-cloud-storage-wuala-is-shutting-down-and-recommends-tresorit.

[14] Z. Li, C. Jin, T. Xu, C. Wilson, Y. Liu, L. Cheng, Y. Liu, Y. Dai, and Z.-L. Zhang. Towards Network-Level Efficiency for Cloud Storage Services. In *Proc. of the ACM Internet Measurement Conference*, 2014.

[15] C. Y. Lin and W. G. Tzeng. Game-theoretic strategy analysis for data reliability management in cloud storage systems. In *Proc. of IEEE Software Security and Reliability*, 2014.

[16] T. Mager, E. Biersack, and P. Michiardi. A measurement study of the wuala on-line storage service. In *Proc. of the IEEE Peer-to-Peer Computing*, 2012.

[17] M. Naldi and L. Mastroeni. Cloud Storage Pricing: A Comparison of Current Practices. In *Proc. of the ACM International Workshop on HotTopiCS*, pages 27–34, 2013.

[18] J. Palviainen and P. P. Rezaei. The next level of user experience of cloud storage services: Supporting collaboration with social features. In *Proc. of IEEE Australasian Software Engineering Conference*, 2015.

[19] M. Rogowsky. Dropbox Is Doing Great, But Maybe Not As Great As We Believed, 2013. http://onforb.es/1Kle8IE.

[20] H. Shen and Z. Li. New bandwidth sharing and pricing policies to achieve a win-win situation for cloud provider and tenants. In *Proc. of the IEEE Infocom*, 2014.

[21] J. Shin, M. Jo, J. Lee, and D. Lee. Strategic management of cloud computing services: Focusing on consumer adoption behavior. *IEEE Transactions on Engineering Management*, 61(3):419–427, 2014.

[22] J. Silber. Shutting down ubuntu one file services, 2015. http://blog.canonical.com/2014/04/02/shutting-down-ubuntu-one-file-services.

[23] D. Stutzbach and R. Rejaie. Understanding churn in peer-to-peer networks. In *Proc. of the ACM SIGCOMM*, 2006.

[24] Z. Wu, M. Butkiewicz, D. Perkins, E. Katz-Bassett, and H. V. Madhyastha. Spanstore: Cost-effective geo-replicated storage spanning multiple cloud services. In *Proc. of the ACM Symposium on Operating Systems Principles*, pages 292–308, 2013.

[25] H. Xu and B. Li. A study of pricing for cloud resources. *SIGMETRICS Perform. Eval. Rev.*, 40(4):3–12, 2013.

[26] H.-S. Yeo, X.-S. Phang, H.-J. Lee, and H. Lim. Leveraging client-side storage techniques for enhanced use of multiple consumer cloud storage services on resource-constrained mobile devices. *Journal of Network and Computer Applications*, 43:142–156, 2014.