Resource Acquisition By Explicit Remuneration For P2P-Based Content Delivery

Joseph C. H. So
The Hong Kong Polytechnic University
and
Po-Choi Wong
The Chinese University of Hong Kong

Many content delivery services are available in the Internet. Such service platforms are built on peer-to-peer (P2P) networks forming Internet clouds. P2P networks provide required resources to provide services with Service Level Agreements. In order to maintain the amount of resources, it is feasible for the service providers soliciting peers' resources contributions by giving explicit remunerations. Naturally, service providers want to minimize the cost of remuneration, as long as the guaranteed services are fulfilled for all users. We consider a model defining the objective function to minimize this cost in service provision and propose a scheme of distributed and cost-efficient resource acquisition. This scheme incorporates the price elasticity of demand, the peer heterogeneity and the limitation of peer information. Our results show that, the SLA fulfillment, the cost efficiency and the load spreading can be enhanced by appropriate resource acquisition schemes even in a heavily loaded network.

Keywords: Peer-to-peer networks, resource acquisition service, content delivery networks

1. INTRODUCTION

Content delivery applications, such as video on demand and data streaming service, are very popular in the Internet. Content providers generally do not own the infrastructure. They rely on the infrastructure service providers to provide low level services, such as computation, storage and bandwidth. The service quality is greatly affected by the availability of resources provided by the underlying networks. These infrastructure providers commit on some quality parameters, such as service availability and downloading bandwidth, so that the service providers can provide some guaranteed services.

Recently, many discussions focus on the realization of Cloud computing [Gilder et al. 2006],[Perry et al. 2008],[Buyya et al. 2008] that makes use of a pool of distributed computing resource as a single point of access for all the computing needs of consumers. Many companies also provide cloud computing infrastructures (e.g. Amazon [Perry et al. 2008], Sun [Sun 2009] and 3Tera [3Tera et al. 2009]). They aim at facilitating application developers to develop, deploy and run applications that are scalable with high service quality. These computing platforms are mainly situated in data centers and the computing resources are located at some server farms. For content delivery services, such provision suffers from the huge amount of bandwidth from data centers and long paths from the content sources to the end users.

Peer-to-peer networks (P2P) provide pools of resources in Internet Clouds. In the Internet Clouds, the computing resources are distributed over the Internet. They provide distributed and programmable network of services across the globe and serve with the resources. P2P networks have tremendous potentials to be used for supporting content distribution services because of its scalability and fast provision of service. Many content delivery applications built on them gain substantial popularity (e.g. PPLive [PPLive 2008], PPStreaming[PPStreaming 2008] and TVants [TVants 2008]).

Unlike most pure P2P network, in which the service user not paying, in commercial offering

Author's address: Joseph So, Hung Hom Bay Campus, The Hong Kong Polytechnic University, Hung Hom, Hong Kong, China

the service users are paying for services with certain level of quality requirement. The requirements are usually stated as some service level agreements (SLA). In order to fulfill SLA, the service provider has to get sufficient resources from the resource contributors over the network. Unfortunately, in P2P networks, the resources are controlled by end-node users. The amount of resources provided by individual peers cannot be determined. Users also have no natural incentives to serve others [Hardin et al. 1968]. Therefore, effective resource acquisition schemes are required to provide the service quality parameters in P2P networks.

Many approaches are proposed to acquire resources in P2P networks. The most prevailing approaches use incentives or penalties to encourage peers to contribute such that peers obtain better service quality for greater resource contribution or punishes free riders by limiting or depriving them from using service [Liu et al. 2007],[Zhang et al. 2004],[Izal et al. 2004],[Cox et al. 2003], [Ngan et al. 2003]. The main drawback of incentive and penalty schemes is that there is still no guarantee what amount of resources are available to each peer. The service quality is provided in the "best effort" manner.

Some content delivery providers, such as PPLive [Huang et al. 2007] and CoolStream [Li et al. 2007], instead of relying solely on the peers' uncommitted and voluntary contributions, try to improve the service quality by adding servers in a service network.

From the view of the peers, what the reward is and its amount are key considerations. Hummel [Wierzbicki et al. 2005] has discussed revenue models and recongised the potential of P2P end-user business model. With the feasibility of micropayment systems (E.g. PayPal [Paypal 2010]), many tiny valued transactions become operatable so that the end user can get revenue as remuneration.

More works consider the resource acquisition as a remunerated service. Some recent works [Ruffo et al. 2007],[Hausheer et al. 2005],[Hefeeda et al. 2003] consider the explicit remuneration for soliciting peers' resource contribution and some [Bocciarelli et al. 2007],[Buyya et al. 2008] consider the Internet as a market of resources that are provided by individual end computers. These works have revealed a new business model, *explicit remuneration in P2P networks*: Peers commit to contribute an amount of resource on each service requested and service providers award peers who contribute their resources to support other peers.

In the generic P2P network, contributions of resources are rewarded by immediate bartering. If a node has no immediate need for such reward (usually the service being provided), it has no incentives to contribute. On the other hand, in explicit remuneration, the contributions are directly remunerated with some tangible rewards, e.g. monetary rewards or credits accumulated for accessing other services in the future. As in model society, currency has replaced bartering as the dominant way of remunerating provision of services. In this model, the paid resource providers are presumed to be fulfilling its commitment as long as they can successfully obtain their remunerations committed by the service user.

Such explicit remuneration strategies can facilitate the fulfillment of SLA in P2P based content distribution services. Figure 1 shows a business model based on explicit remuneration. This model supports three kinds of peers in a network. The first type of peers contribute and consume resources at the same time. The second type of peers contribute resources, and get tangible rewards. The third type focus on consuming resources, and agree to pay instead. They are willing to pay instead of sharing their resources, as they might have physical limitations (e.g. behind firewalls), or have other concerns (e.g. security).

It is worth to note that in generic P2P networks, peers belong to the first type. From the service provider perspective, this simple bartering service from resource providers cannot guarantee the sufficiency of resources for all users, especially the paid peers, who are the main revenue source.

Explicit remuneration incurs an additional operating cost. P2P service providers need to determine the cost implication for providing a resource acquisition (RA) service with SLA. As far as we know, few studies address the cost implication during resource acquisition in explicit remuneration. In our previous work [So et al. 2008], we have proposed a model to analyze the cost involved in resource acquisition. In [So et al. 2009], we further investigate the effect of high

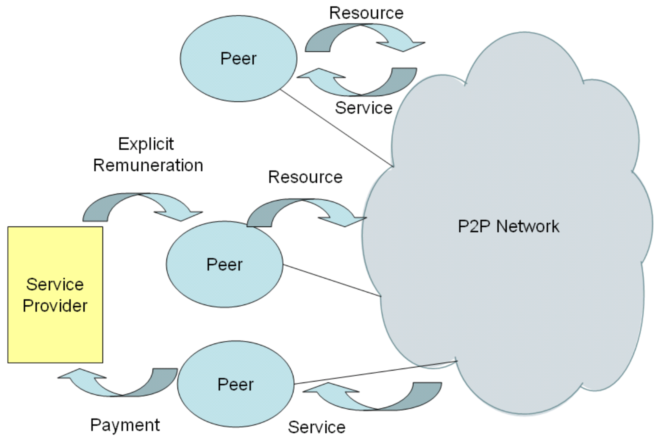


Figure. 1: Explicit remunerated peer and service consuming peers

load environment. In this work, we extend our work to consider the SLA fulfillment, proximity awareness and the load balancing capability in content delivery networks and the adaptation on the Service-Oriented Architecture (SOA) framework.

The resources available in a peer are limited. The acquisition scheme should be able to allocate resource efficiently and effectively so that the services are provided with desired quality. It has to cater for different network topologies and changes in the available resources of the peers. We refer this problem as a distributed resource allocation problem. The research questions being addressed are:

(1) In a P2P system with no central agent that can have all system parameters, can we develop an efficient algorithm to acquire sufficient amount of resources to satisfy the SLA commitment of all user applications in a completely distributed manner?

(2) In addition to satisfying the application resource requirements, how can these schemes be optimized to enhance some specific system performance parameters (e.g. the cost and the fairness of load distribution)?

This paper considers the problem of resource acquisition and its cost implication to the service provider. We look into the behavior of a P2P network with explicit remuneration by investigating their resource acquisition cost under various load distribution and network topologies. These schemes can explore the available resources and provide every user with sufficient resources to satisfy the QoS requirement while the cost due to remuneration is kept to a low level. It performs well in the environment that peers have only local information and heterogeneous capabilities.

This paper is structured as follows. We first define the framework of the resource acquisition in the service-oriented architecture, review the model of a P2P network with cost parameters on resources and define the problem being tackled in section 2. We present our heuristic and solutions in section 3. The evaluation results are discussed in section 4 and the related works are reviewed in section 5, and we conclude in the last section.

2. SERVICE MODEL AND RESOURCE ACQUISITION

We propose a framework that unifies the resource acquisition for multi-application P2P network in the resource acquisition. The framework distinguishes the functional roles of a node in the Internet cloud in the resource acquisition. Newly added P2P applications can use the resources provided by the service offered in the underlying virtual machine.

Our model aims at facilitating the service provision in building Service-Oriented Architecture [Zhang et al. 2007]. As shown in Figure 2, adopting a similar model in SOA, there are three roles in the P2P based resource acquisition service: Resource provider, Resource consumer and Resource registry. A resource provider publishes its available resources to the resource registry. These computers are looking for remuneration for their contributions. A resource requestor queries the resource registry for an interested resource and obtains the location information of the corresponding resource provider. Then, the resource requestor reserve resources from and binds to the resource provider. The peers' resources are billed like the "paid-per-usage" for resources in utility computing. One key issue we need to emphasize is that, unlike the central index server used in the earliest P2P implementation like Napster, in contemporary P2P environment, the resources are published in a distributed manner, so the registry is distributed instead of centralized.

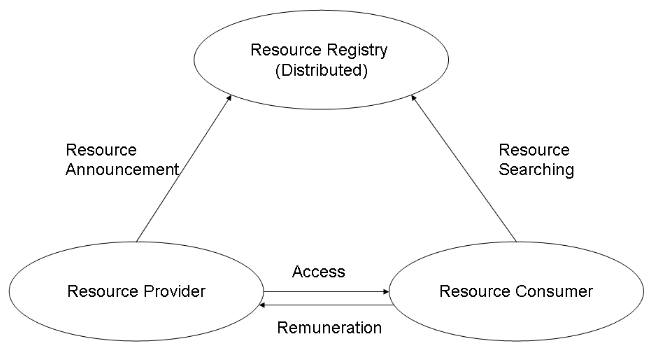


Figure. 2: Operation model of Resource Acquisition with Explicit Remuneration

2.1 Resource Acquisition

Resource Acquisition (RA) service aims at reserving sufficient resources to satisfy the upper layer SLA requirements for all service users. The storage space and the uploading bandwidth provided by a peer are considered as the main resources acquired in many content delivery services, such as online backup (e.g. Amazon's Simple Storage Service (S3) [Amazon 2009],[Junqueira et al. 2005]) and video-on-demand services ([Hsieh et al. 2004]). Therefore, in this paper, guarantees in storage availability and data rate for every user are considered as the main QoS parameters provided by the RA service.

We consider the provision of content delivery applications in a P2P network as illustrated in Figure 3. A resource user (or simply a *user*) is an application initiated by a peer in the network using the network resources. A user (e.g. u_1) arrives, it will register with a peer (P11), which becomes its *root peer*. When a user requests for a RA service with designated amount of storage space and bandwidth, (20GB, 500Kbps), the RA service provider will acquire the resources available from the overlay network. Each peer has committed a prescribed set of resources to be shared and available for other peers to reserve. (E.g. P8 has 5GB storage space and 300Kbps outgoing bandwidth available, while P4 has 6GB and 500Kbps). Upon receiving requests from a root peer, they will reserve unreserved portions of the committed resources to these users. Each peer charges RA service provider and the unit charge is different in each peer and varies with the peer condition. The RA service provides will choose among them.

After registration and deliver network formation, a user is ready to deliver a broadcast session. In addition, a micropayment system (e.g. PayPal [Paypal 2010]) has to be deployed. The user and the chosen resource providing peers register with this system, which is preferably a trusted

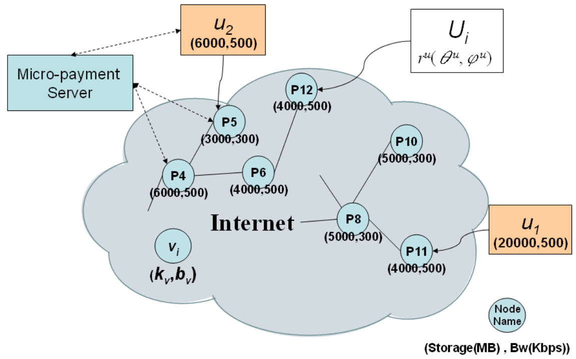


Figure. 3: Service Provision of Resource Acquisition

third party with no favorite on both sides. During the broadcast session, the user will pay the servicing peers periodically. In other words, the servicing peers get remunerated immediated every certain period of time and it can check with the payment server to see whether the user is fulfilling the payment commitment. The user will also check lower-tier peers of a provider to check to provision of resource by their upper-tier peers. If a peer is not providing the committed resources, the user will halt the payment immediately, and the user will invite another provider to take up the role so that the service will sustain.

As the resources from each peer are limited, an efficient resource acquisition scheme is needed to determine the amount of resources that should be allocated to each user from each peer in the network, so that the RA service can satisfy the SLA requirements of every user. It has to cater for the changes in the available network resources. We refer this problem as a **distributed resource allocation** (DRA) problem.

2.2 Problem formulation

We first summarize the formulated P2P model and the key parameters concerned as presented in [So et al. 2008]. The connectivity structure of a P2P network is considered as an undirected graph with nodes as the vertices and the links between them as edges. Suppose we have a graph $G = (V, E)$, where $|V| = n$ is the number of nodes and $|E| = m$ is the number of links. There are $|U|$ content providers $U = \{u_1, u_2, \ldots, u_t\}$. Each peer node v has two parameters to describe its commitment of resources sharing: storage space (k_v) and outgoing bandwidth (b_v). k_v^u and b_v^u define the partition of storage space and the uploading bandwidth allocated to a user $u \in U$.

We aim at minimizing the cost of resource reservation while maintaining the amount of resources of the system capable of serving all users' requests. End-to-end SLA requirement of users' demand is converted into local resource requirement at each node.

Service demand: Each user u has the requirement describing vector defining the resource demand (θ^u, ϕ^u) where θ^u is storage space and ϕ^u is bandwidth.

Decision variables: For the set of registered resource provider nodes P, the set of nodes acting as the candidates of resource providers for user u is denoted by P^u. The set of decision variables are defined as follows: For each peer v, $\{k_v^u\}_{u \in U}$ and $\{b_v^u\}_{u \in U}$ are the two vector of numeric decision variables, representing the storage space and bandwidth reserved for each user u on peer v respectively.

With this model, we formulate the problem statement: *Given a network G(V,E), a set of registered provider nodes with the storage capacities $\{k_v\}_{v \in P}$ and the outgoing bandwidth $\{b_v\}_{v \in P}$, their cost per unit storage $q_v(k)$ and cost per unit bandwidth $q_v(b)$, serving a set of users U, each with the resource demand of (θ^u, ϕ^u). The serving peers will be remunerated based on the amount of resource reserved. With the requirement that all users are given with sufficient resource in good proximity, we find a set of resource provider nodes $\{P^u\}_{u \in U}$, a set of resource allocation on the* *nodes* $\{k_v^u, b_v^u\}_{v \in P}$ *such that the overall cost due to remuneration is minimized.*

Mathematically, the optimization problem can be formulated as finding the values of k_v^u and b_v^u that satisfy the following objective:

Minimize:

$$\sum_{u \in U} \sum_{v \in P^u} (k_v^u q_v(k_v^u) + b_v^u q_v(b_v^u)) \tag{1}$$

Subject to

$$\theta^u \leqslant \sum_{v \in P^u} k_v^u , \quad \forall u \in U \tag{2}$$

$$\phi^u \leqslant \sum_{v \in P^u} b_v^u , \quad \forall u \in U \tag{3}$$

$$|k_v| \geqslant \sum_{u \in U} k_v^u , \quad \forall v \in P^u \tag{4}$$

$$|b_v| \geqslant \sum_{u \in U} b_v^u , \quad \forall v \in P^u \tag{5}$$

The cost is the sum of two components: the first term refers to the total cost of resource for the fulfillment of the requirement capacity, and another one is the cost incurred by the services requiring bandwidth guarantee. $q_v(k)$ and $q_v(b)$ are functions of the storage and bandwidth of the provider nodes and the load on that node. We will elaborate it in later sections.

The constaints in (2) and (3) define the user requirements in terms of storage capacity and different bandwidth. The constraints in (4) and (5) guarantee the provider nodes have sufficient capacity in terms of storage and bandwidth to satisfy the requirement.

Further, in order to preserve the bandwidth for the required storage capacity, the size of bandwidth reserved is proportional to the amount of the storage reserved. We define the bandwidth storage ratio (bsr) as b_u/k_u. That is $b_v^u \geqslant \frac{k_v^u}{k^u} b^u , \quad \forall v \in P^u, \forall u \in U$ (The overhead in establishing connections to multiple peers is negligible.) We assume that the aggregated amount of resources available in the network is adequate to satisfy the total requirement of the all users. Therefore, $\sum_{u \in U} \theta^u \leqslant \sum_{v \in P^u} k_v$ and $\sum_{u \in U} \phi^u \leqslant \sum_{v \in P^u} b_v$.

To ensure good proximity of resource, the search will be limited to a scope within a specific hop count. Therefore, the number of hops from a resource provider to a peer limited within a specific hop count $MAX_H C$.

Similar to many proposed overlay network construction mechanisms [Zhu et al. 2003],[Liu et al. 2008], nodes in the network are assumed to be on live in the period of service. Nodes are obedient to their commitment. Once committed, resources providing nodes are assumed to make its committed resources available throughout the whole service period. It is also applicable to the overlay network constructed by stable set-top-box (STB) [Cha et al. 2008],[Liu et al. 2008]. The effect of peer churning and dynamics is considered in the future work.

3. SOLUTION AND HEURISTICS

The objective of the solution is to acquire sufficient amount of resources to satisfy the application requirement in an optimal cost effective way, where the cost is the value of remuneration. We will explain our proposed scheme to achieve it from the perspectives of *resource identification, resource reservation, and cost model.* The considered factors include the interaction of four key factors: the load assignment, the cost model, the allowable bandwidth provided by each provider, and the capability allocation on each provider.

3.1 Design Heuristics

Our design is inspired by two key aspects in service provision: the feasibility of solution in SLA fulfillment and the cost of implementation.

In a distributed system with a large number of nodes, it is generally believed that obtaining the global knowledge and distributing it to all peers is not feasible. In a reasonably large network, each peer only has information about a part of the network. The information includes the knowledge of network topology, online peers and their resources. The scope is confined by the distance between the peers. Therefore, we are particularly interested in an efficient method that can generate near-optimal solutions for this distributed environment and heuristic approachs for solving the problem.

Similar to many P2P systems, such as PPlive and PPStream, after a peer has joined the system, it will obtain a list of peers from a bootstrap node as its start-up neighbors. It is assumed that peer can communicate directly with their direct neighbors and obtain information about the other peers from them and changes its neighbor list according to the peer dynamics.

It is natural that the resource users attempt to look for low cost supply of resources. In a free market of resources provision in P2P-based Internet Clouds, peers can offer the unit price of resources they would like to provide. Simply speaking, we materialize an economic concept of *price elasticity of demand* (PED) [Case et al. 1999]: the demand quantity is sensitive to the price. It is used as the basis in designing a remuneration scheme such that when a user chooses a peer as a resource provider due to its relatively low charge, the chosen peer is also lightly loaded. Eventually, the difference of loading among peers is regulated to be more balanced, and the number of providers serving each user can also be regulated.

In this regard, we consider a *proximity-aware* and *cost-aware* scheme for resource discovery and resource reservation. First, the scheme identifies a group of candidate peers that have potential to provide the required resource. These peers are within the minimum number of hop count for the user. Second, we design a cost-aware resource reservation scheme for retrieving sufficient resources to satisfy the resource requirement of all users. In contrast to centralized resource allocation algorithms, both schemes only require local knowledge about their neighboring nodes. To realize this idea, we consider three design aspects of a resource acquisition scheme:

(1) Resource Identification: Each user searches for appropriate resource providers. The resource providers close to the user are recognized as the high priority provider.

(2) Resource Reservation: For each user, the amount of resources retrieved can fulfill the SLA requirement and the involved cost is minimized among all the identified providers.

(3) Remuneration Model: It reflects the price elasticity of demand. The demand for the resource on a particular provider is regulated to prevent overloading. In other words, the load on each peer is allocated fairly so that heavily load peers charge for higher cost to prevent overloading.

In the scheme, peers are nodes on the network and act as both clients and servers in terms of resource reservation requests and fulfillment. As in many P2P systems, we assume that each peer is able to perform the following peer management functions: 1) Network Connectivity Management: It manages neighbor information and keeps track of the available resources in their neighbors. It probes its neighbors periodically for their liveliness, resource availability and neighborhood. The resource is released and available immediately when the user is offline or terminates a service. 2) User Registration: The peer that the user is working with will manage the user registration. 3) Resource Control: Each peer manages the resources allocated to each user from other peers and it would also keep track of its registered users.

Apart from the above mentioned assumptions,our scheme requires that the user node responses fast enough for coming and leaving of nodes. Only peers with good reputation can register as resource providers. Selection of reputable peers have widely discussed in many other works [Wierzbicki et al. 2005],[Adler et al. 2005],[Chun et al. 2005],[Shrivastava et al. 2005]. Furthermore, we also have the following assumptions: 1) The intermediate network nodes and links would not incur any bottleneck. 2) A peer only handles single request at a time and the resource will not be locked up due to the transition period of a request processing.

3.2 Initialization and Resource Identification

The resource acquisition algorithm is described in this and the next sub-sections. It consists of three procedures: initialization, resource identification and resource reservation.

3.2.1 *Initialization.* Both peers and users need initialization. When a peer joins the system, it will contact some well known bootstrapping nodes $v_{bootstrap}$ which maintain a short list of peers that are currently alive. ($v_{bootstrap}$ acts as a part of the resource registry mentioned in Section 2.) The new peer probes this list of peers for their liveliness, resource availability and neighborhood. Afterwards, the peer manages neighborhood information and keeps track of the available resources in its neighbors.

A user initialization starts when a new user u_{new} joins the system via a peer, which then becomes the user root peer v^u_{root}. v^u_{root} contacts a well known bootstrap peer $v_{bootstrap}$ to obtain its peerID and the list of its neighbors h_v. v^u_{root} will further contact h_v to obtain their list of neighbors h^2_v. By joining h_v and h^2_v, it forms a set of initial *candidate peers* h_c for resource provision (i.e. peers in two hops away). Therefore, each v_{root} will act as a resource registry for the neighbors within two-hops.

3.2.2 *Resource Identification.* This procedure aims at identifying the availability of resources in the peers around the v^u_{root}. v^u_{root} generates a call-for-resource (CFR) and sends it to each peer in h_c.

Each CFR consists of the following fields: Initiator ID (IID): A field uniquely identifying the initiating peer v^u_{root}; User ID (UID): A field uniquely identifies a user; CFR ID (CID): A field uniquely identifies the message from the same initiator; User Requirement Descriptor (L^u): The required capacity for satisfying the user requirement on the storage space and bandwidth.

When a peer receives CFR, if it has spare storage space and spare bandwidth, it will tentatively reserve the corresponding resource and reply to v^u_{root}. The reply message states the peer remaining storage space and bandwidth (k^r_v, b^r_v). The responding peer v will also reply with the list of its neighbors with their spare storage space and bandwidth $v.h_v.k_v$ and $v.h_v.b_v$. The response also contains the cost of the required resource. It is noted that some nodes may receive the same CFR from several different nodes. Peers will ignore duplicating CFR, which is identified by IID within a fixed validity period.

v^u_{root} gathers responses from all peers in h_c and evaluates the available resource. v^u_{root} waits the sufficient responses so that the targeted amount of resource is reached. If the amount is not sufficient after obtaining all responses or after the MAXIMUM_WAITING_TIME, v^u_{root} will initiate the next round of resource searching. The scope of searching is expanded by neighbor list learnt from h_c. Peers with no response after timeout are considered unreachable. For each round of scope expansion, the distance of new candidates is increased. The MAX_HC will limit the distance of the provider from the root peer so that the latency due to hop count will be reduced. MAX_HC is application-specific and its determination is regarded as a future work.

3.3 Resource Reservation

In this phase, v^u_{root} selects the optimal composition of resource providers and informs the candidate peers on the actual amount of resource to be reserved. After the resource identification phase, v^u_{root} knows the available resources from h_c as well as the unit cost of resources $q_v(k^r_v)$ and $q_v(b^r_v)$. v^u_{root} will determine the amount of resources should be acquired from each peer in h_c and generate a list of the *provider peers* p_u with a tuple of information (CID, $(k^u_v,\ b^u_v)$) where (k^u_v, b^u_v) is User Acquisition Descriptor (L^u_v) stating the amount of storage space and bandwidth reserved for a particular user from a peer v. When a peer v receives L^u_v, it allocates a set of resource (k^u_v, b^u_v) to the user u to fulfill the request.

We consider two schemes of resource reservation: *Greedy Breath First algorithm* (GBFA) and *Cost-aware Requisition algorithm* (CRA). We developed CRA with the consideration that the resources acquired should be of the least cost.

Greedy Breath First algorithm (GBFA). Greedy resource searching is a typical approach used in many P2P networks (e.g. Fasttrack and Gnutella). In GBFA, a peer will contribute its available capacity for each CFR. As long as v^u_{root} finds some resources that can be reserved, it will reserve them immediately by sending reserveResource(L^u_v) to this peer. The candidate will response with reserveConfirm message. v^u_{root} then recalculates the remaining capacity. We define a *bandwidth-to-storage ratio* (bsr) as the ratio of bandwidth size to storage size in the user requirement.

Algorithm 1 GBFA

Get h_v from $v_{bootstrap}$
$h_c \leftarrow h_v$, $k^r_u \leftarrow \{\}$, $b^r_u \leftarrow \{\}$, $\theta^u_e \leftarrow \theta^u$, $\phi^u_e \leftarrow \phi^u$
foreach v in h_c
    if (sumOf(k^u_v) $\geqslant \theta^u$ or sumOf(b^u_v) $\geqslant \phi^u$)
        break
    endif
    SendCFR(v, CFR(IID, CID, UID, TTL, L^u, θ^u_e, ϕ^u_e))
    Wait for responses from v for MAXIMUM_WAINING_TIME
    Get(v, (k^r_v, b^r_v, $Q(\theta^u)$, $Q(\phi^u)$, $h_c.ngb$))
    if ($k^r_v > \theta^u_e$ and $b^r_v > \phi^u_e$), $k^u_v \leftarrow \theta^u_e$ and $b^u_v \leftarrow \phi^u_e$
        else if ($k^r_v : \theta^u_e < b^r_v : \phi^u_e$), $k^u_v \leftarrow k^r_v$, $b^u_v \leftarrow \phi^u k^r_v / \theta^u$
            else $k^u_v \leftarrow \theta^u b^r_v / \phi^u$ and $b^u_v \leftarrow b^r_v$
    endif
    SendReserveResource(v, CID, L^u_v)
    GetReserveConfirm(v, CID, (k^u_v, b^u_v))
    $\theta^u_e \leftarrow (\theta^u_e - k^u_v), \phi^u_e \leftarrow (\phi^u_e - b^u_v)$
    Add $h^c.ngb$ to h^c
endfor

If a candidate peer has sufficient resources to satisfy the storage and bandwidth requirement ($k^r_v > \theta^u_e$ and $b^r_v > \phi^u_e$), it will satisfy the response immediately, and set $k^r_v = \theta^u_e$ and $b^r_v = \phi^u_e$. If not, the peer uses its all available resources to satisfy the requirement as far as possible. If the short in the storage amount is greater than that of the bandwidth ($k^r_v : \theta^u_e < b^r_v : \phi^u_e$), the storage of k^r_v and the bandwidth of $b^u_v \leftarrow \phi^u k^r_v / \theta^u$ will be allocated to user u; otherwise, the bandwidth of b^r_v and the storage of $k^u_v \leftarrow \theta^u b^r_v / \phi^u$ will be allocated to user u.

For each GetReserveResource(L^u_v) sent out, v^u_{root} will wait for acknowledgement of successful reservation with actual amount (k^u_v, b^u_v). If v^u_{root} cannot receive acknowledgement within the timeout period, it will assume the target peer is unreachable and try the next peer.

This algorithm is expected to be in favor to the applications requiring small number of hops from the user. However, it tends to use up quickly the resources nearby a user requesting for a great amount of resources.

Cost-aware Requisition algorithm (CRA). In this algorithm, we propose to introduce some margin in the process of reservation, so that more candidates can be selected as providers in order to facilitate the optimization in the provision cost.

In this algorithm, after obtaining response from a peer, the root peer, instead of acquiring as much resource as possible immediately in GBFA, it will wait for more peers responding until the total amount of resource available in the candidate list solicited has a *margin* (m) exceeding the user requirement. This margin allows the root peer to evaluate the L^v_u of each peer in h^c, so that the set of peers with lowest unit cost ($q_v(k^u_v) + bsr \times q_v(b^u_v)$) will be selected. $reserveSource(k^u_v, b^u_v)$ will be sent to peers that are selected and the decline message will be sent to peers not selected. Hopefully, the root peer will choose the set of offers from the peers that cost the minimum. The cost is calculated in accordance with the current load on a peer and the requested amount as described above.

Algorithm 2 CRA

1. Get h_v from $v_{bootstrap}$
2. $h_c \leftarrow h_v$
3. $\theta^u_m \leftarrow \theta^u \times (1+m)$, $\phi^u_m \leftarrow \phi^u \times (1+m)$
4. foreach v in h_c
5.   if (sumOf(k^r_v) $\geqslant \theta^u_m$ or sumOf(b^r_v) $\geqslant \phi^u_m$)
6.   break
7.   endif
8.   SendCFR(v, CFR(IID, UID, CID, TTL, L^u, θ^u, ϕ^u))
9. endfor
10. Wait for responses from v until (sumOf(k^u_v) $\geqslant \theta^u_m$ and sumOf(b^u_v) $\geqslant \phi^u_m$) or MAXIMUM_WAINING_TIME
11. foreach v response
12.   Get(v, (k^r_v, b^r_v, $Q(\theta^u)$, $Q(\phi^u)$, $h_c.ngb$))
13.   Add $h^c.ngb$ to h^c
14. endfor
15. Sort $v_{candidate}$ by the unit cost
16. Determine $v_{pp} \in v_{candidate}$ s.t. $min(q_v(k^u_v) + bsr \times q_v(b^u_v))$
17. foreach v_{pp} in $v_{candidate}$
18.   if (sumOf(k^u_v) $\geqslant \theta^u$ and sumOf(b^u_v) $\geqslant \phi^u$
19.   break
20.   endif
21.   SendReserveResource(v_{pp}, (k^u_v, b^u_v))
22.   GetReserveConfirm(v_{pp}, (k^u_v, b^u_v))
23. $\theta^u_e \leftarrow (\theta^u_e - k^u_v), \phi^u_e \leftarrow (\phi^u_e - b^u_v)$
24. endfor

3.4 Cost Determination

The unit cost of storage $q_v(k^r_v)$ and bandwidth $q_v(b^r_v)$ are two distinct types of costs referring to the unit cost of the storage space with specific capacity and that of certain guarantee in the uploading bandwidth.

We employ a *Load aware (LA) model*. In this model, it follows the principle that the resource in a peer is scarcer, it will become more costly. The unit cost of storage $q_v(k^r_v)$ and bandwidth $q_v(b^r_v)$ will increase respectively when the available amount decreases. They follow the expression of $F_k + G_k(k_v - k^r_v)$ and $F_b + G_b(b_v - b^r_v)$, where F_k and F_b are the fixed unit costs of an instant of a storage reservation and a bandwidth reservation and G_k and G_b are the unit costs varying with the storage availability and the bandwidth availability.

Therefore, the cost for storage and bandwidth for a user is the product of the unit cost and the amount of storage reservation, that is, $Q_v(k^u_v, k^r_v) = k^u_v \times [F_k + G_k(k_v - k^r_v + k^u_v/2)]$; similarly, the cost for bandwidth for a user is $Q_v(b^u_v, b^r_v) = b^u_v \times [F_b + G_b(b_v - b^r_v + b^u_v/2)]$.

4. MEASUREMENT AND SIMULATION

In order to evaluate the algorithms in a systematic way, four performance metrics are investigated: SLA fulfillment, service cost, load spreading and proximity of providers. *SLA fulfillment* indicates the capability of the scheme of resource acquisition in SLA fulfillment in different load situation. *Service cost* is the target of the optimization and shows the effectiveness of algorithms'

Table I: Parameters used in experiment

Symbol	Description	Default values / {Allowed values}
P	No. of Peers	5000
U	No. of Users	50-2000
Deg	Ave.Peer Deg.	4
T	Topology	Random / {Superpeer, PowerLaw, Random}
C	Cost Model	(G_k, G_b, F_k, F_b) = (0.01,1,1,1)
k_v	Storage/peer	$Pr(k_v > k) = (k/k_m)^{-\alpha}$, k_m=200, α=0.5
b_v	Bandwidth/peer	$Pr(b_v > b) = (b/b_m)^{-\beta}$, b_m=500, β=1.0
k^u	Storage/user	Gaussian dist. {500 - 2000}MB
b^u	Bandwidth/user	Gaussian dist. {500 - 2000}Kbps

cost sensitivity. *Load spreading* indicates the ability to prevent some nodes from overloading. *Proximity of providers* measures the capability of the searching nearby resource in order to reduce the searching time in resource acquisition. We have performed a set of simulations and investigated the behavior of unstructured P2P networks.

We designed a simulator that implements the mechanism explained in this paper. We simulate a P2P network with 5×10^3 peers and the number of users increases from zero to a number such that the amount of resource requested exceeds the amount of resource in the system. We intentionally increase the load to maximum handling capacity to observe the performance under heavy load. From each peer, the amount of storage shared is in Pareto distribution $Pr(k_v > k) = (k/k_m)^{-\alpha}$ with $k_m = 200$ in terms of MB and α is 0.5 and the amount of uploading bandwidth shared is in Pareto distribution $Pr(b_v > b) = (b/b_m)^{-\beta}$ with $b_m = 500$ in terms of Kbps and β is 1.0. For each user, the request for resources is ranging from 500 to 2000MB in terms of storage and 500 to 2000Kbps in terms of bandwidth both in Gaussian distribution. Other parameters being used are listed in Table I.

4.1 SLA Fulfillment

The ability of the scheme in acquiring sufficient resource for satisfying SLA is investigated first. As shown in Figure 4, we found that, while in the low load situation (the system utilization less than 75%), for both GBFA and CRA, all users could successfully acquire the requested resources. It shows that when the loading increases, as long as the system is not heavily loaded, the explicit remuneration scheme is able to fulfill the required SLA.

However, under the heavily loaded situation (the system utilization > 75% but < 100%), CRA outperforms GBFA as CRA can maintain the SLA fulfillment percentage above 95%. The improvement in CRA is attributed to its ability in directing additional load on a highly loaded to some lightly loaded peers. It helps in spreading out the load over the network, and the chance of a peer being overloaded and refuses to take new requests is lowered. The chance of nearby nodes get overloaded is lowered. They can satisfy more requests.

4.2 Service Cost

The total cost measures how cost efficient the system is in resource acquisition. We use load aware (LA) cost model to compare the cost efficiency of CRA and GFBA. The unit cost of each 100MB of storage space is evaluated using the following expression: $[Q(k_u) + Q(b_u)]/k_u$ where k_u is the size of storage space in unit of 100MB. As shown in Figure 5, the average unit cost increases with the number of users. There are three observations from the perspective on CRA effectiveness in cost reduction, the cost optimization of CRA and the response in excessive load.

First, when the load is set low or modestly high (the system utilization is below 75%), the CRA has significant advantage over GBFA. There is cost reduction of 10% to 15% for CRA(m=0.5) and 18% to 35% for CRA(m=1.0). It verifies that the CRA can successfully reduce the average

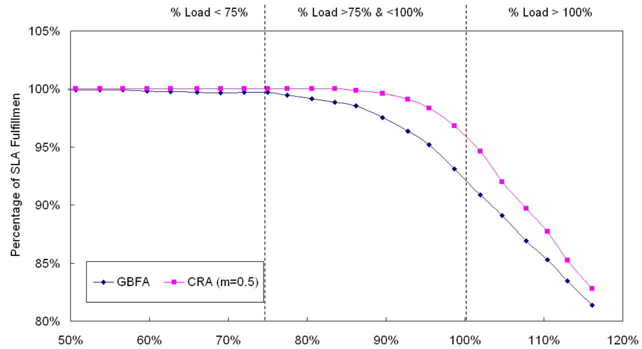


Figure. 4: Percentage of Request with SLA Fulfilled

unit cost of resource acquisition.

Second, in CRA when m increases, the cost is reduced for the same amount of loading. However, the effect diminishes when it is large enough, as we can see that the unit cost for m=2 and m=4 is about the same in all load. As we can see that when $m \to \infty$, CRA is virtually searching for all available peers in the system. That is, it performs global optimization on the unit cost before choosing peers as the resource providers. This result shows that while our design heuristic only aim at near-optimal cost optimization, the very close optimal solution can be obtained in CRA with small margin (m).

Third, when the system is nearly or already overloaded, the unit cost of all peers becomes the same. It is expected as when the system utilization comes to the full, all resources are used up regardless on their prices. However, when we compare it with Figure 4, we can see with the same unit cost, CRA can successfully maintain better SLA fulfillment capability.

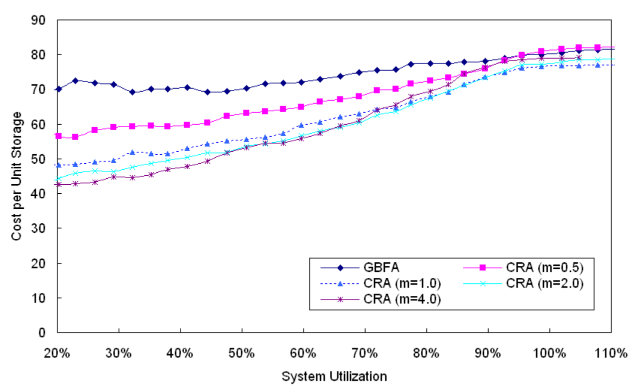


Figure. 5: Cost of storage unit of GBFA vs CRA algorithms

4.3 Load Spreading

The occupancy of the storage of each peer is defined as the ratio of storage space being reserved to the amount of the storage, $Occup(k_v) = \frac{\sum_{u \in U} k_v^u}{k_v}$. The standard deviation of storage occupancy reflects the deviation of the loading on peers in the system, that is

$$\sigma(Occup) = \sqrt{\frac{1}{n-1}\sum_{v\in P}[Occup(k_v) - \overline{Occup(k_v)}]^2} \quad (6)$$

where $\overline{Occup(k_v)}$ is the average $Occup(k_v)$ of all peers. As shown in Figure 6, in the case of low load (No. of users < 500), CRA with low margin (m=0.5 and m=1.0) has basically the same $\sigma(Occup)$ as GBFA. The load balancing capability is the same. while for a low margin (m=0.5 and m=1.0), $\sigma(Occup)$ is kept lower than GBFA. It shows that, the fairness of load distribution in the system will not be affected in CFA if the margin is not too large. When the margin increases, $\sigma(Occup)$ also increases. When the system utilization is low, the need for the load balancing is less stringent as the resource available is not affected by the load distribution. Instead, the goal of cost reduction is more significant.

When the load is high (No. of users > 800), the increase in margin will lead to a decrease in $\sigma(Occup)$. The reason for the reduction may reflect the CRA ability in not choosing peers with high unit cost due to their heavily load taken. The load balance capability can lead to the improvement in SLA fulfillment as shown in Figure 4.

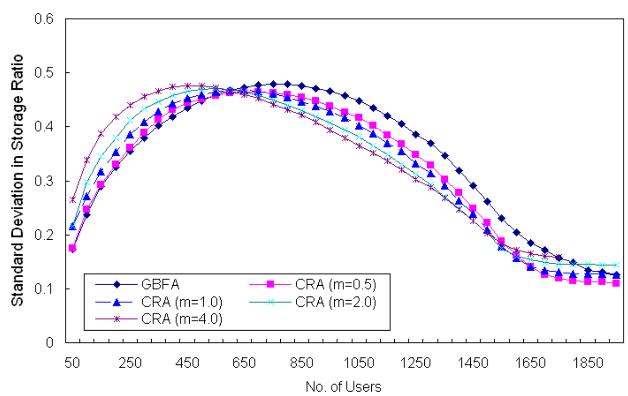


Figure. 6: Average Deivation in storage occupancy of peer storage vs. no. of users

4.4 Proximity of Providers

Lookup path length is the number of hops needed to look up before each instance of resource reservation on a provider is carried out. The lookup path should involve as few nodes as possible such that the look up is efficient. We set a MAX_HC on the dilation of the search scope in the network, which is the maximum path length. When we compare the hop count, we can see from Figure 7, both GBFA and CRA can keep the number of hops in range of 1.3 to 2.5. Furthermore, when the margin increases, the number of hops also increases. It demonstrates, the proximity aware CRA and GBFA are both successful in searching for resource within a few number of hops.

5. RELATED WORKS

In content delivery networks (CDN), contents are delivered by dedicated servers[Xu et al. 2004],[Held et al. 2006]. Servers are placed in some strategic locations. As the requirement of time-to-market is going more tight, the high cost and long deployment period of servers setup in the designated location hinder its usage in the application which requires fast deployment. A new paradigm of service-oriented network is emerging. This paradigm emphasizes on the fast service deployment and service customization for end users. The required resources that are required to support the services can be provided in the on-demand basis.

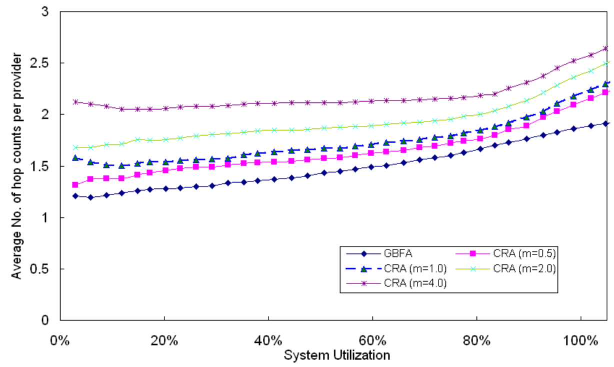


Figure. 7: Average number of hops per provider for each user

The operation model of SOA consists of three roles: *Service provider, Service requestor, and Service Registry* [Zhang et al. 2007]. Recently, more works have proposed how to incorporate P2P network service in the SOA model [Zhou et al. 2008],[Bocciarelli et al. 2007]. In the current SOA implementation, the registry is implemented as a centralized agent and service can be published and searched by Web Service Description Language (WSDL) and Universal Description, Discovery, and Integration (UDDI). We adopt this model in the resource acquisition and customize it and adapt to the P2P network, in which resources are distributed across the network with no single central point of control.

To be service-oriented, the resource availability in P2P network becomes a significant concern. As found in some empirical studies of a video system [Huang et al. 2007],[Li et al. 2007], the aggregate upload resource in a P2P content delivery network is about 2.5 times of the aggregate user demand. Therefore, it is reasonable to assume that the aggregate resources available in the system are sufficient to cater for the requirements of all users.

In the context of resource acquisition in an overlay network, from the literature, different approaches were proposed: *incentives and penalties*, *servers placement* and *remuneration*.

Using incentives and penalties, peers obtain better service quality for their contributions [Liu et al. 2007],[Zhang et al. 2004],[Izal et al. 2004],[Cox et al. 2003]. In [Liu et al. 2007], layered videos are provided to users with better video quality for their greater uploading bandwidth contribution. In PeerCast[Zhang et al. 2004], a node that contributes more bandwidth will have a better position in the identifier space, so that, it will experience eventually a lower delay. BitTorrent [Izal et al. 2004] implements Tit-for-tac and choking mechanism which temporarily refuses a node to download if it is not uploading to another peer. A storage network, Samara [Cox et al. 2003] punishes uncooperative nodes that delete or lose others' data by deleting those nodes' data.

Instead of relying solely on peers' contribution in form of bartering, PPLive [Huang et al. 2007] and CoolStream [Li et al. 2007] enhance the overall video streaming quality by pre-locating some servers in certain strategic locations. Peer-assisted VoD [Huang et al. 2007] further highlights the significance of servers in stable bandwidth injection to the system. This type of approach can significantly enhance the service quality to the peers, but it requires a greater capital investment and a longer time for planning and deployment.

More works attempt to solicit resources by means of *explicit remuneration*. For examples, PeerMart [Hausheer et al. 2005] considers the use of markets for trading resources in P2P systems. [Ruffo et al. 2007] uses a profit-sharing strategy for content distribution services. [Figueiredo et al. 2005] proposed to use monetary payments to enhance the peer availability for service provision.

6. CONCLUSION

In this paper, we have investigated a peer resource acquisition mechanism in terms of service availability as well as the cost minimization scheme. We consider a scheme that identifies the appropriate peers in the system and determines the amount of resource to be reserved according to the peers' characteristics, the remuneration function and the current load. We make use of the idea that the unit cost increases when the amount of unoccupied resource on each peer decreases. When compared with GBFA, the cost is reduced by CRA. Our simulation results show that, especially in a heavily loaded network, CRA can successfully acquire sufficient resource for all users while the cost is substantially lower than the simple greedy scheme GBFA. The results show that a well designed peer resource acquisition scheme with consideration on remuneration for the peer contribution not only improves the cost efficiency, but also enhances the fulfillment of the service quality commitment.

REFERENCES

3Tera. 3tera, Oct 2009. http://www.3tera.com/.

Adler, M., Kumar, R., Ross, K., Rubenstein, D., Suel, T., and Yao, D. Optimal peer selection for P2P downloading and streaming. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE 3* (March 2005), 1538–1549 vol. 3.

Amazon. Amazon web service, Oct 2009. http://aws.amazon.com/.

Bocciarelli, P., D.Ambrogio, A., and Angelaccio, M. QShare: QoS-enabled description and discovery of services in soa-based P2P applications. *E*nabling Technologies, IEEE International Workshops on 0 (2007), 159–166.

Buyya, R., Yeo, C. S., and Venugopal, S. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. *H*igh Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference on (Sept. 2008), 5–13.

Case, K. E., and Fair, R. C. *P*rinciples of Economics, 5th ed. Prentice-Hall, 1999.

Cha, M., Rodriguez, P., Moon, S., and Crowcroft, J. On next-generation telco-managed P2P tv architectures. In *I*PTPS'08 (Florida, 2008 2008).

Chun, B. G., Zhao, B. Y., and Kubiatowicz, J. D. Impact of neighbor selection on performance and resilience of structured P2P networks. In *I*n Proceedings of 4th International Workshop on Peer-to-Peer Systems IPTPS (2005).

Cox, L. P., and Noble, B. D. Samsara: honor among thieves in peer-to-peer storage. In *S*OSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles (New York, NY, USA, 2003), ACM, pp. 120–132.

Figueiredo, D., Shapiro, J., and Towsley, D. Incentives to promote availability in peer-to-peer anonymity systems. In *I*CNP '05: Proceedings of the 13TH IEEE International Conference on Network Protocols (Washington, DC, USA, 2005), IEEE Computer Society, pp. 110–121.

Gilder, G. The information factories, Oct 2006. http://www.wired.com/.

Hardin, G. The tragedy of the commons. *S*cience 162 (1968), 1243–1248.

Hausheer, D., and Stiller, B. Peermart: The technology for a distributed auction-based market for peer-to-peer services. In *4*0th IEEE International Conference on Communications (ICC 2005) (Seoul, Korea, May 2005).

Hefeeda, M. M., Habib, A., and Bhargava, B. K. Cost-profit analysis of a peer-to-peer media streaming architecture. Tech. rep., Purdue University, 2003.

Held, G. *A* practical guide to content delivery networks. Auerbach Publications, Boca Raton, FL, 2006.

Hsieh, H. Y., and Sivakumar, R. Accelerating peer-to-peer networks for video streaming using multipoint-to-point communication. *I*EEE Communications Magazine 42, 8 (Aug. 2004), 111–119.

Huang, C., Li, J., and Ross, K. W. Can internet video-on-demand be profitable? In *S*IGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications (New York, NY, USA, 2007), ACM, pp. 133–144.

Huang, G. Experiences with PPLive. In *P*eer-to-Peer Streaming and IP-TV Workshop, Sigcomm (2007), ACM.

Izal, M., Urvoy-Keller, G., Biersack, E., Felber, P., Hamra, A. A., and Garces-Erice, L. Dissecting BitTorrent: Five months in a torrent's lifetime. In *P*roceedings of Passive and Active Network Measurement, 5th International Workshop (2004), vol. 3015, Springer, pp. 1–11.

Junqueira, F., Bhagwan, R., Hevia, A., Marzullo, K., and Voelker, G. M. Surviving internet catastrophes. In *A*TEC '05: Proceedings of the annual conference on USENIX Annual Technical Conference (Berkeley, CA, USA, 2005), USENIX Association, pp. 4–4.

Li, B., Xie, S., Keung, G. Y., Liu, J., Stoica, I., Zhang, H., and Zhang, X. An empirical study of the CoolStreaming+ system. *IEEE Journal on Selected Areas in Communications 25*, 9 (2007), 1627–1639.

Liu, S., Zhang-Shen, R., Jiang, W., Rexford, J., and Chiang, M. Performance bounds for peer-assisted live streaming. In *SIGMETRICS '08: Proceedings of the 2008 ACM SIGMETRICS international conference on Measurement and modeling of computer systems* (New York, NY, USA, 2008), ACM, pp. 313–324.

Liu, Z., Shen, Y., Panwar, S. S., Ross, K. W., and Wang, Y. Using layered video to provide incentives in P2P live streaming. In *P2P-TV '07: Proceedings of the 2007 workshop on Peer-to-peer streaming and IP-TV* (New York, NY, USA, 2007), ACM, pp. 311–316.

Ngan, T. W. J., Wallach, D. S., and Druschel, P. Enforcing fair sharing of peer-to-peer resources, Feb. 2003.

PalPal. PalPal, 2010. http://www.paypal.com/.

Perry, G. How cloud and utility computing are different, Feb 2008. http://gigaom.com/2008/02/28/how-cloud-utility-computing-are-different/.

PPLive. PPlive, 2008. http://www.pplive.com/.

PPStreaming. PPStreaming, 2008. http://www.ppstreaming.com/.

Ruffo, G., and Schifanella, R. Fairpeers: Efficient profit sharing in fair peer-to-peer market places. *J. Netw. Syst. Manage. 15*, 3 (2007), 355–382.

Shrivastava, V., and Banerjee, S. Natural selection in peer-to-peer streaming: from the cathedral to the bazaar. In *NOSSDAV '05: Proceedings of the international workshop on Network and operating systems support for digital audio and video* (New York, NY, USA, 2005), ACM, pp. 93–98.

So, J. C. H., and Wong, P. C. Cost-aware resource acquisition for guaranteed service in P2P networks. *Services Part II, IEEE Congress on 0* (2008), 183–188.

So, J. C. H., and Wong, P. C. Remuneration aware resource acquisition in P2P overlay. *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE* (Jan. 2009), 1–5.

Sun. Sun's network, Oct 2009. http://network.com/.

TVants. TVants, 2008. http://www.tvants.com/.

Wierzbicki, A., and Goworek, K. Peer-to-peer direct sales. *Peer-to-Peer Computing, 2005. P2P 2005. Fifth IEEE International Conference on* (Aug.-2 Sept. 2005), 106–113.

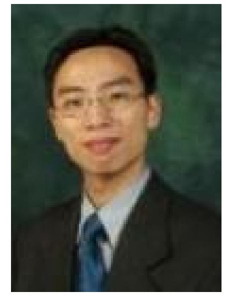
Xu, D., Kulkarni, S. S., Rosenberg, C., and keung Chai, H. A CDN-P2P hybrid architecture for cost-effective streaming media distribution. *Computer Networks 44* (2004), 353–382.

Zhang, J., Liu, L., Pu, C., and Ammar, M. H. Reliable end system multicasting with a heterogeneous overlay network. Tech. rep., Georgia Tech [http://etd.library.gatech.edu/dspace-oai/request] (United States), 2004.

Zhang, L. J., Zhang, J., and Cai, H. *Services Computing: Core Enabling Technology of the Modern Services Industry*. Tsinghua University Press, 2007.

Zhou, J., Rautiainen, M., and Ylianttila, M. Community coordinated multimedia: Converging content-driven and service-driven models. *Multimedia and Expo, 2008 IEEE International Conference on* (April 2008), 365–368.

Zhu, C., Liu, Z., Zhang, W., Xiao, W., and Yang, D. Analysis on greedy-search based service location in P2P service grid. In *P2P '03: Proceedings of the 3rd International Conference on Peer-to-Peer Computing* (Washington, DC, USA, 2003), IEEE Computer Society, p. 110.



Dr Joseph So graduated from The University of Hong Kong with a Bachelor's degree in Electrical and Electronic Engineering. He then obtained a Master of Science from The Hong Kong Polytechnic University and a PhD in Information Engineering from The Chinese University of Hong Kong. Dr So had been working as an Internet Protocol Specialist at Pacific Century CyberWorks (PCCW) for several years, concentrating on Broadband Network Development, Network Management Systems, and Data Centre Services. Dr So also have been the Network Operation Centre Manager of a multinational internet infrastructure service provider. He is a Chartered Engineer. Dr So is particularly interested in the Peer-to-peer Networks, Internet and World Wide Web Technologies, Communication Networks, and Educational Technologies.

Prof. P.C. Wong is a renowned IT expert in Hong Kong. In 1998, he launched the Hong Kong Cyber Campus project to bring the Internet era to schools in Hong Kong. In 2004, he launched the VChina project, a video conferencing network linking hundreds of schools in HK, China, and overseas. He published widely and has received many awards, the IT Application Award in 1998, two US Patents and the IT Excellence Award in 1999, two Grand Linux Awards in 2003, and the 10th anniversary QEF Outstanding Project Award in 2008. In 2001, he received the Medal of Honor from Hong Kong for his contributions to the use of IT in education. Prof. Wong was a professor in the Chinese University of Hong Kong for twenty years before changing to Hang Seng Management College to become their Associate Vice President (Development).