Cost-Profit Analysis of a Peer-to-Peer Media Streaming Architecture*

Mohamed M. Hefeeda, Ahsan Habib, and Bharat K. Bhargava
CERIAS and Department of Computer Sciences
Purdue University
West Lafayette, IN 47907
{mhefeeda, habib, bb}@cs.purdue.edu

Abstract

We study the economic aspects of P2P systems. We present a cost-profit analysis of a media streaming service deployed over a peer-to-peer (P2P) infrastructure. We consider the limited capacity as well as the heterogeneity of peers in the analysis. The analysis shows that with the appropriate incentives for participating peers, the service provider achieves more profit. In addition, the analysis shows how the service provider can maximize its revenue by controlling the amount of incentives offered to peers.

By comparing the economics of P2P and conventional client/server media streaming architectures, we show that with a relatively small initial investment, the P2P architecture can realize a large-scale media streaming service.

1 Introduction

Cost is a crucial factor for the success of a large-scale media streaming service. It is important for both the supplier and the consumer of the service. In [6], we propose a novel peer-to-peer (P2P) media streaming architecture that scales well to a large number of clients. In this paper, we show that the P2P architecture is cost effective for the supplier and the consumer. The supplier does not need powerful servers and caches/proxies with high bandwidth network connectivity to serve the clients. Clients get low cost service by sharing some of their extra, often under-utilized, resources such as bandwidth and storage.

Current media streaming architectures are mainly composed of a streaming entity and a set of requesting clients. The streaming entity could be a single server, or a set of servers and/or proxies. This entity is

---

*This research is sponsored in part by the National Science Foundation grants CCR-001712 and CCR-001788, CERIAS, and IBM SUR grant

responsible for providing the requested media files to *all* clients. The number of concurrent clients such a system can support is limited by the resources of the streaming entity. The limitation mainly stems from the out bound network bandwidth, but it could also be due to the processing power, memory size, or the I/O speed of the server. The current approaches have limitations in reliability and scalability. The reliability concern arises because only one entity is feeding all clients. The scalability of these approaches is not on a par with the requirements of a media distribution service that spans an Internet-scale number of potential users. Adding more users requires adding a commensurate amount of resources to the supplying servers. We generically refer to all variations of the aforementioned approach as the *conventional* architecture for media streaming.

1.1 P2P Media Streaming Architecture

We summarize the important aspects of the P2P architecture in the following and refer the interested reader to [6] for the details. The basic idea of our approach is shown in Figure 1. In the P2P architecture, a peer may act as a client and/or as a mini-server. As a client, it requests media files from the system. A peer may opt to store segments of the media files that it has already consumed for a specific period of time. As a mini-server, it can provide these segments to other requesting peers in the system. Although individually each of these mini-servers adds only a little to the overall system capacity, combining a large number of them can significantly amplify the capacity of the system. Peers join the system along with their resources. As more peers join the system, the system capacity increases and this leads to a scalable system that can potentially support many clients. The P2P architecture consists of the following entities.

1. **Peers.** This is the set of clients' machines that are currently participating in the system. We denote $\mathbb{P} = \{P_1, P_2, \cdots, P_{\mathcal{N}}\}$ as the set of all peers in the system. Every peer P_i controls its level of cooperation with the system through three parameters: (1) R_i (in Kb/s), the maximum rate peer P_i is willing to share with others; (2) G_i (in bytes), the maximum storage space the peer is willing to allocate to store segments of one or more media files; and (3) C_i, the maximum number of concurrent connections that can be opened to serve requesting peers.

2. **Seeding server(s).** One or a subset of the peers that seeds the new media files into the system. In a *commercial* media streaming service, these seeding servers will typically be owned by a service provider. We intentionally chose the name *seeding* (not streaming) servers to indicate that their main functionality is to *initiate* the streaming service and not to serve all clients at all times. These seeding "servers" do not negate the P2P nature of the model, in which peers help each other in getting the work done, i.e., streaming the media files.

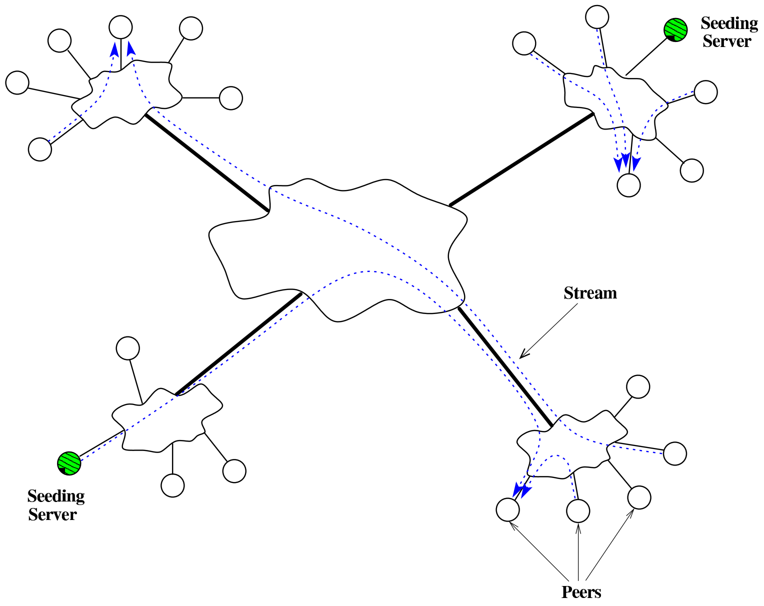

Figure 1: P2P architecture for media streaming.

3. **Stream.** A stream is a time-ordered sequence of packets belonging to a specific media file.
4. **Media files.** The set of movies currently available in the system. Every movie is characterized by a size in bytes, and is recorded at a constant bit rate R Kb/s. A media file is divided equally into N segments. A segment is the minimum unit a peer can cache. A supplying peer may provide the cached copy of the segment at a rate lower than the required rate R. Therefore, in general, one segment can be streamed to the requesting peer from multiple peers at the same time. According to our protocol (see [6]), every peer will supply a different *piece* of the segment proportional to its streaming rate.

1.2 Paper Contributions and Organization

The paper studies the economic issues of a media streaming service deployed over a P2P architecture. The objective is three fold. First, we show that by offering the appropriate incentives for the participating peers, the media provider can gain more revenue. Second, we show that with a relatively small initial investment in the basic infrastructure, the P2P architecture can realize a large-scale media streaming service. Third, we verify the claim that the P2P architecture is more cost effective and more profitable than the conventional architecture.

The remainder of the paper is organized as follows. Section 2 describes the proposed cost-profit model. Section 3 shows how a service provider induces peers to cooperate and how this cooperation affects the provider's revenue. The revenue maximization is presented in Section 4. Section 5 compares the profit for the P2P architecture and the conventional architecture. Section 6 reviews the related work and Section 7 concludes the paper and sketches future extensions.

2 Cost-Profit Model

In our cost-profit model we have two parties: a media streaming service provider and a community of peers (clients) who are interested in the service. We assume that the provider will deploy a small set of *seeding* servers with a limited capacity. This initial investment in addition to the running cost constitute the total cost imposed on the provider. The provider earns a fixed amount of v dollars for each client that successfully receives the entire movie. v is called revenue per customer.

A peer first joins the system as a client, i.e., it requests a movie and pays for it[1]. Then, this peer may become a supplying peer by caching and streaming parts of the movie. A supplying peer incurs two types of cost: storage and bandwidth. Since media files are large and consume significant bandwidth to stream, the cost imposed on the supplying peer is non-trivial. We call this cost a *disutility* or negative utility. Unless there is a *reward* or a positive utility that offsets this disutility, peers will typically not cooperate. (In [1], it was shown that about 70% of users share nothing with the Gnutella network.)

To motivate peers to cooperate, we assume that the provider grants (to peers) αv dollars ($0 \leq \alpha < 1$) for every customer served by those peers in the system. We call α the *inducement factor*. A higher α will induce more cooperation from peers. Every contributing peer gets a *share* of the revenue proportional to its contribution. For example, if peer P_x serves the entire request, i.e., provides all the N segments at the full rate R, it will get αv dollars. Whereas if P_x serves only $N_x, 0 \leq N_x \leq N$, segments at rate $R_x, 0 \leq R_x \leq R$, P_x gets $(R_x/R)(N_x/N)\alpha v$ dollars.

The analysis in the following sections considers the case of one movie with a duration of T hours. For this movie, we compute the provider's revenue and cost over a period of $h \geq T$ hours. Table 1 lists the symbols used in the analysis.

3 Incentives for Peers and Revenue for the Provider

We study how the cooperative peers can contribute to the total revenue of the provider, if they get the appropriate incentive. In the conventional architecture, all clients receive the movie from the centralized server

---

[1] In general, the system may attract some peers to contribute without being former customers. For example, some small companies have relatively powerful machines, which they do not fully utilize. Given the current economic hardship, these companies may find it worthwhile to share the unused resources. Our analysis is still valid in this case.

Table 1: Symbols used in the analysis.

Symbol	Description
A	Average Number of copies of the movie cached by peers
C	Total fixed cost incurred by the provider (in \$)
F	Total profit made by the provider (in \$)
h	Analysis period (in hours)
l	Number of customers
m	Maximum number of concurrent customers that can be served by the server
N_x	Number of segments cached by peer P_x
u_x	Fraction of time peer P_x is online
R_x	Rate at which peer P_x streams
T	Duration of the movie (in hours)
t	Time variable (in hours)
V	Total revenue (in \$)
v	Revenue per customer (in \$/customer)
α	Inducement factor
γ	Depreciation factor

(or its proxies/caches, if any). Therefore, the *maximum* total revenue the provider can make is limited by the capacity of the server and its proxies. To compute the maximum total revenue, we assume sufficient customer requests that keep the server busy. We define m_{conv} to be the maximum number of concurrent customers that can be served by the server and its proxies. Since every streaming session lasts for T hours, the maximum number of customers that can be satisfied per hour is m_{conv}/T. Therefore, the maximum total revenue V_{conv} during the period h is given by:

$$V_{conv} = \frac{h}{T}\, m_{conv}\, v. \tag{1}$$

In the P2P architecture, clients get portions (possibly all) of the media file from other peers. This allows for serving more customers with the same server capacity. Let us define A to be the average number of copies cached by the set of peers $\mathbb{P}$ in the system. A can be computed as:

$$A = \sum_{x=1}^{|\mathbb{P}|} \frac{R_x}{R}\,\frac{N_x}{N}\, u_x. \tag{2}$$

The summation in Equation (2) computes the effective number of copies that are made available through peers' contributions. It takes into considerations the limited capacity and heterogeneity of peers, since peers do not offer server-like behavior as shown in [11]. The equation accounts for two facts. First, peers are not always online through the term u_x, which is the fraction of time peer P_x is available. Second, peers do not cache all segments at the full rate through the term R_xN_x/RN. Dividing A by T results in the number of requests that can be served per hour, since every request takes T hours to stream. Multiplying the result by

h gives the total additional number of customers l_{add} that can be served by peers during the period h. Thus, l_{add} is computed as:

$$l_{add} = \frac{h}{T} A. \tag{3}$$

From the provider's perspective, l_{add} is the total number of additional customers (beyond the seeding servers capacity) that can be served due to contributions by peers. Each additional customer adds $(1-\alpha)v$ dollars to the total revenue of the provider. We define m_{p2p} to be the maximum number of concurrent customers that can be served by the seeding servers (typically $m_{p2p} \ll m_{conv}$). The maximum total revenue in the P2P architecture for the period h is given by:

$$V_{p2p} = \frac{h}{T} m_{p2p} v + l_{add} (1-\alpha) v. \tag{4}$$

The maximum total revenue is not limited by the maximum capacity m_{p2p} of the seeding servers. Rather, it grows in proportion to l_{add}, since $\alpha < 1$.

Two questions arise: How do we get peers to cache A copies? How long will it take? Intuitively, A is positively correlated with the inducement factor α and the time t. Larger α values motivate either more peers to cache, or the same peers to cache more segments, or both. This increases the effective number of cached copies of the movie. For a given α, the system gets a specific caching rate from peers. Thus, the number of cached copies accumulates over the time. Generally speaking, A can be expressed as a function of α and t, i.e.,

$$A = f(\alpha, t). \tag{5}$$

The function f can be estimated (and refined over time) through statistics gathered from customers. It depends on how peers react to the incentives offered by the provider. To illustrate the analysis procedure, we consider a simple *hypothetical* function:

$$A = f(\alpha, t) = k\, \alpha^{\frac{1}{\epsilon_1}}\, t^{\frac{1}{\epsilon_2}} \tag{6}$$

where k, ϵ_1, and ϵ_2 are positive constants greater than one. Notice that, ϵ_1 and ϵ_2 are unitless. The unit of k should be chosen to balance the equation; it depends on the value of ϵ_2. These constants are used to control the shape of the function f.

We plot Equation (6) for $k = 100$, $\epsilon_1 = 2$, $\epsilon_2 = 3$, and several values of α in Figure 2. Equation (6) and Figure 2 show the expected positive correlation between A and both α and t. They also indicate that the number of cached copies A will saturate after a reasonably long time, which is intuitive because we have a finite number of peers with finite capacities. We believe that the shape of a realistic $f(\alpha, t)$ (i.e., one that is

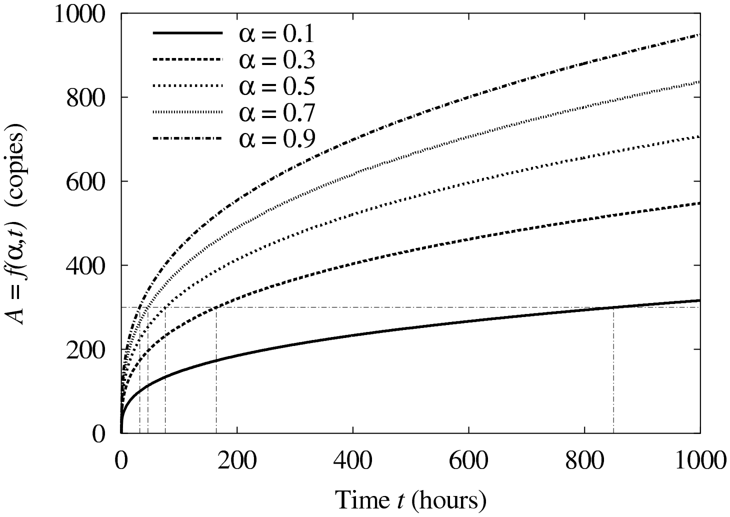

Figure 2: The effect of the inducement factor α on the average number of cached copies A over a period of time t.

fitted to data gathered from customers over a long period of time) will look somewhat similar to the curves shown in Figure 2. We are unable to verify this conjuncture due to lack of data.

Figure 2 can be used to answer the two questions as follows. Suppose that the provider expects a total of 30,000 client requests distributed over a period of 20 days. Assume a movie of a duration $T = 2$ hours and an average active period per day of 10 hours. Thus, $h = 20 \times 10 = 200$ hours. If we ignore the limited capacity of the seeding servers, the provider needs $l_{add} = 30,000$ customers to be served by peers. Thus, from Equation (3), A needs to be 300 by the end of the 200-hour period. In Figure 2, we see that the minimum value of α that can achieve the expected capacity within the 200-hour period is 0.3. Taking $\alpha = 0.1$ will take about 850 hours to get the required capacity. On the other hand, a value of 0.7 for α will quickly motivate peers, and the required capacity will be obtained in less than 46 hours. However, as Equation (4) indicates, $\alpha = 0.7$ will yield a smaller revenue for the provider than $\alpha = 0.3$.

In summary, using a sample function $f(\alpha, t)$, we showed how the provider can determine the appropriate inducement factor to motivate peers to cache enough copies of the media file for the expected number of customers within a specific target period.

4 Revenue Maximization

Suppose that the provider expects many customers that will absorb any offered capacity in the system within a period of h hours. This can happen in the case of releasing a popular movie. Given the estimated $f(\alpha, t)$,

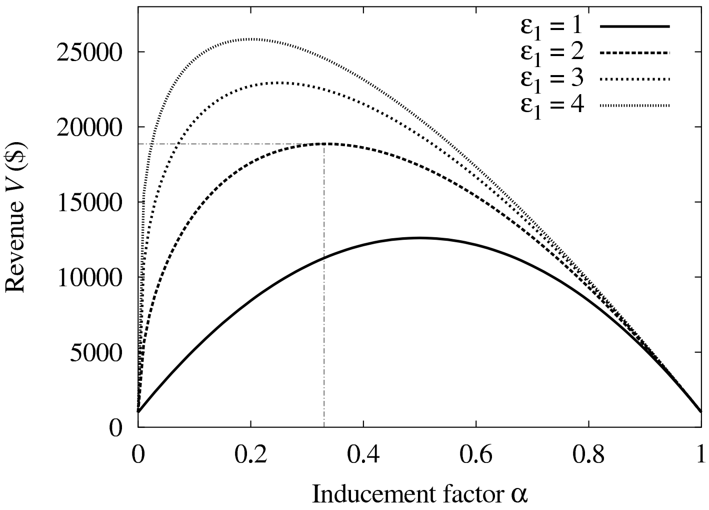

Figure 3: Finding the optimal α to maximize the provider's revenue

we can find the optimal inducement factor α_{opt} that maximizes the total revenue for the provider. In this case, t is fixed and equals h, since we maximize the total revenue over the entire period h. If we substitute $f(\alpha, h)$ in Equation (4), we get the revenue V_{p2p} as a function of only one unknown, α. As an example, consider our hypothetical $f(\alpha, t)$ as described above. After substitution, Equation (4) reduces to:

$$V_{p2p} = \frac{h}{T}\, m_{p2p}\, v + k\, \alpha^{\frac{1}{\epsilon_1}}\, h^{\frac{1}{\epsilon_2}}\, \frac{h}{T}\, (1-\alpha)\, v. \tag{7}$$

Differentiating (7) w.r.t. α and equating the result with zero, we get the optimal inducement factor α_{opt}:

$$\alpha_{opt} = \frac{1}{(1+\epsilon_1)}. \tag{8}$$

Figure 3 shows the relation between V_{p2p} and α for $\epsilon_1 = 1, 2, 3$, and 4. The figure shows the importance of choosing the correct inducement factor to maximize the provider's revenue. For instance, if $\epsilon_1 = 2$, the optimal inducement factor α_{opt} is 0.33 and it achieves the optimal total revenue of $18,865. In contrast, for $\alpha = 0.1$, the provider achieves a total revenue of $14,210, and for $\alpha = 0.6$, achieves a total revenue of $15,381.

5 Profit Comparison

This section compares the cost and profit for the P2P architecture and the conventional architecture. The comparison is approximate and ignores several practical factors. Nevertheless, it gives some guidelines for the feasibility of deploying a large-scale media service deployed over a P2P infrastructure.

We first define the net profit made by a provider within a period of h hours as:

$$F = V - C, \tag{9}$$

where V is the total revenue obtained by serving customers during the period h, and C is the total cost incurred by the provider during that period. C considers only the fixed cost, because the running cost is implicitly considered in computing V. The running cost is considered in the revenue per customer v. The fixed cost is the cost for deploying the whole infrastructure. A reasonable assumption to make is that the fixed cost is proportional to the maximum number of customers that can be served concurrently by the server. To serve more customers, more powerful machines and larger network bandwidth are needed. Since we analyze the net profit over a period of h hours, we consider the cost over the same period. Thus, the total cost within a period of h hours (for both architectures) can be approximated as:

$$C = \gamma\, m \tag{10}$$

where γ is the proportional factor. γ accounts for the fact that the cost incurred within the period h is a fraction of the total cost needed to deploy the infrastructure. We may call γ the *depreciation* factor, which is proportional to the decrease in the monetary value of the machines due to their use for h hours.

To compute the maximum net profit, we need to compute the total number of customers l that can be served during the period h. For the conventional architecture, customers can be served as long as the server's capacity allows. We define l_{conv} as the maximum number of customers that can be served by the server in h hours. It is computed as:

$$l_{conv} = \frac{h}{T}\, m_{conv}. \tag{11}$$

The net profit as a function of the number of customers l for the conventional architecture can be computed as:

$$F_{conv} = \begin{cases} l\,v - C_{conv}, & 0 \le l \le l_{conv} \\ l_{conv}\,v - C_{conv}, & l > l_{conv} \end{cases} \tag{12}$$

Similarly, the maximum number of customers that can be served by the seeding server l_{p2p} in the P2P architecture is given by:

$$l_{p2p} = \frac{h}{T}\, m_{p2p}. \tag{13}$$

Besides these l_{p2p} customers, the P2P architecture can serve additional customers due to peers contributions. For each additional customer, the provider gains $(1-\alpha)v$ dollars. For simplicity, we assume that the additional capacity added by peers will be used only when the seeding server runs out of capacity. Therefore, the net profit for the P2P architecture can be computed as:

$$F_{p2p} = \begin{cases} l\,v - C_{p2p}, & 0 \le l \le l_{p2p} \\ l_{p2p}\,v - C_{p2p} + & \\ (l - l_{p2p})\,(1-\alpha)\,v, & l > l_{p2p} \end{cases} \tag{14}$$

To illustrate this comparison, let us consider the following values: $T = h = 1$, $v = 1$, $\gamma = 0.1$, $m_{conv} = 1000$, and $m_{p2p} = 0.01 \times m_{conv} = 10$ $(m_{p2p} \ll m_{conv})$. Hence, $l_{conv} = 1000$, $l_{p2p} = 10$, $C_{conv} = 0.1 \times 1000 = 100$, and $C_{p2p} = 0.1 \times 10 = 1$. Using these values, Equations (12) and (14) are reduced to:

$$F_{conv} = \begin{cases} l - 100, & 0 \le l \le 1000 \\ 900, & l > 1000 \end{cases} \tag{15}$$

$$F_{p2p} = \begin{cases} l - 1, & 0 \le l \le 10 \\ 9 + (1-\alpha)\,(l-10), & l > 10 \end{cases} \tag{16}$$

In Figure 4, we show the net profit for the conventional architecture and the P2P architecture with $\alpha = 0.2, 0.4$, and 0.6. The figure shows that with a much less investment in the basic infrastructure (one hundredth), the P2P architecture is able to achieve a comparable profit with a moderate number of customers. Moreover, the P2P architecture achieves much higher profit as the number of customers increases. Even when $\alpha = 0.6$ (meaning that peers are getting a larger portion of the per-customer profit than the provider), the P2P architecture still achieves a higher profit for the provider for $l \ge 2225$. Finally, the conventional architecture has negative profit (or loss) for $l < 100$. In contrast, the P2P architecture has negative profit for only $l < 1$. This is a desirable property, especially for small and start up companies.

6 Related Work

There has been a significant research effort on P2P systems, mainly tackling the searching problem, see for example [9], [10], [12], and [13], and recently the replication problem [4]. However, the economic aspects of peer-to-peer systems have received less attention in the literature.

Peer-to-peer systems suffer from a serious *free riding* problem as shown in [1] (through measurements study) and in [5] (through game-theoretic analysis). Free riding means that peers are provided with no incentives to contribute resources into the system. Therefore, only small fraction of peers contribute to the growth of the system. Since the free riding problem has detrimental effects on the future of P2P systems, we as well as authors of [5] and [1] advocate using incentive mechanisms to ensure a healthy growth of such systems. In [5], the authors construct a game theoretic model for P2P systems and study the equilibria of user strategies

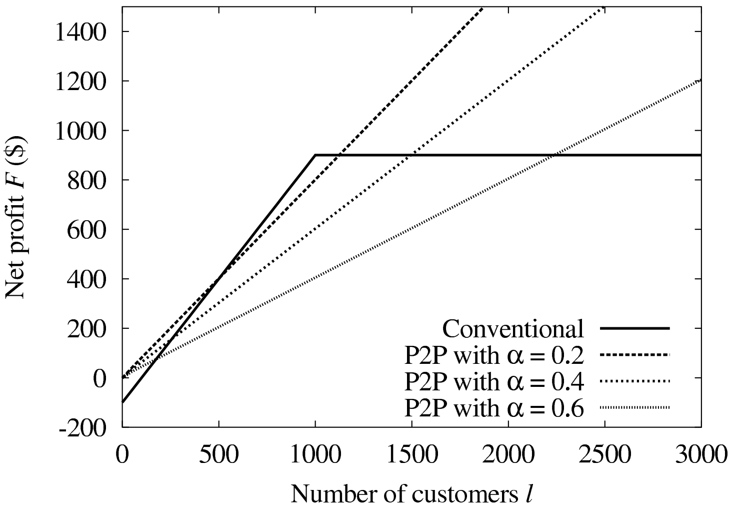

Figure 4: Profit comparison between conventional and P2P architectures.

under different payment mechanisms such as micro-payment, points-based, and rewards for sharing. Economic incentives are proposed in [7] to motivate users to keep the content within the subscription community, instead of relying on tamper resistance protocols. Incentive mechanisms are also used to desynchronize users' resource requests in order to avoid congestion in the network [8]. By probabilistically waiving the cost of using resources in a chosen time slot, the authors show (using game theory) that the load can be de-focused.

Finally, the cost comparisons presented in [2] and [3] show that a distributed architecture for video on-demand services has a lower cost than a centralized one. This is similar to our effort. Our analysis shows the potential of a P2P architecture for a higher cost effectiveness compared to the distributed approach, because the P2P architecture does not require powerful caches or proxies. It needs, however, limited-capacity seeding servers for a short time, which cost much less than powerful caches.

7 Conclusions and Future Work

We presented an approximate cost model and showed the economic potential of a large-scale media streaming service built on top of a P2P infrastructure. We showed that using the inducement factor, the provider can manage the level of cooperation offered by peers. Specifically, our analysis can be used in two ways. First, if we have sufficiently large number of customers in a given period, the *optimal* value of the inducement factor α_{opt} that maximizes the provider's revenue can be computed. Second, if there is a limit on the expected number of customers, the *appropriate* value of α can be chosen to ensure the sufficient capacity within the target period.

We compared the profit that obtained in the P2P architecture and the conventional client/server architecture. Our comparison showed that even with large inducement factors (i.e., peers get a larger portion of the per-customer profit than the provider), the P2P architecture still achieves a higher profit with a reasonable number of customers. The comparison also demonstrated the cost-effectiveness of the P2P architecture, since it requires a small initial investment in the infrastructure.

The main extension of this work is to consider the case of multiple movies. Specifically, the relationship between the inducement factor and the relative popularities of the movies needs to be studied and included into the analysis. Finally, obtaining a more realistic data on peers' reaction to the offered inducement factor will substantiate the analysis. This can be done by obtaining a survey from a random sample of customers.

References

[1] E. Adar and B. Huberman. Free riding on Gnutella. *First Monday*, 5(10), October 2000.

[2] S. Barnett and G. Anido. A cost comparison of distributed and centralized approaches to video-on-demand. *IEEE Journal on Selected Areas in Communications*, 14(6):pp. 1173–1183, August 1996.

[3] S. Chan and F. Tobagi. Distributed servers architecture for networked video services. *IEEE Transactions on Networking*, 9(2):125–136, April 2001.

[4] E. Choen and S. Shenker. Replication strategies in unstructured peer-to-peer networks. In *Proc. of ACM SIGCOMM'02*, pages 177–190, Pittsburgh, PA, USA, August 2002.

[5] P. Golle, K. Leylton-Brown, and I. Mironov. Incentives for sharing in peer-to-peer networks. In *Proc. of The Second workshop on Electronic Commerce (WELCOM'01)*, Heidelberg, Germany, November 2001.

[6] M. Hefeeda, B. Bhargava, and D. Yau. On-demand media streaming over the internet. Technical report, CERIAS TR 2002-20, Purdue University, June 2002.

[7] B. Horne, B. Pinkas, and T. Sander. Escrow services and incentives in peer-to-peer networks. In *Proc. of The ACM Conference on Electronic Commerce (EC'01)*, Tampa, Florida, October 2001.

[8] K. Leylton-Brown, R. Porter, B. Prabhakar, Y. Shoham, and S. Venkataraman. Incentives mechanisms for smoothing out a focused demand for network resources. In *Proc. of The ACM Conference on Electronic Commerce (EC'01)*, Tampa, Florida, October 2001.

[9] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proc. of ACM SIGCOMM'01*, San Diago, CA, USA, August 2001.

[10] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*, November 2001.

[11] S. Saroiu, P. Gummadi, and S. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proc. of Multimedia Computing and Networking (MMCN02)*, San Jose, CA, USA, January 2002.

[12] I. Soitca, R. Morris, M. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proc. of ACM SIGCOMM'01*, San Diago, CA, USA, August 2001.

[13] B. Yang and H. Garcia-Molina. Efficient search in peer-to-peer networks. In *Proc. of ICDCS'02*, Vienna, Austria, July 2002.