A Framework for Cost-Effective Peer-to-Peer Content Distribution

Mohamed M. Hefeeda
Department of Computer Sciences
Purdue University, West Lafayette, IN 47907
mhefeeda@cs.purdue.edu

1. INTRODUCTION

Motivated by the success of the peer-to-peer (P2P) paradigm in the last few years and by the immense number of the often underutilized end systems connected to the Internet, we propose a collaborative P2P framework for cost-effective content distribution. We focus mainly on the problem of streaming large media files (e.g., movies) to a large-scale user community. The key idea of our approach is: instead of deploying powerful caches at many locations, the P2P model relies on resource contributions from peers (client machines). Every peer may contribute a little, but there is an enormous number of them. As peers contribute resources, the overall system capacity increases and more clients can be served. By properly *motivating* peers to share some of their resources, the system achieves significant cost-effectiveness. The P2P approach strives to push the contents even closer to the clients: contents are obtained from fellow peers within the *same* network domain. This potentially can lead to a good performance and a substantial decrease in the network load.

The proposed P2P framework can be used in two settings. First, it can serve as an *infrastructure* through which content providers disseminate contents to clients by employing and aggregating resources from participating peers [3]. In this case, content providers should motivate peers to contribute resources to the system. Second, it can be used as a *cooperative* resource-sharing environment, in which peers cooperate and coordinate among themselves to serve requests from other peers [5, 6]. In this case, incentive mechanisms should be developed to ensure fair contribution and consumption of resources. The P2P framework has the potential to create a substantial value in a cost effective manner. However, realizing this potential requires addressing a number of challenging research tasks, including: (1) How to select and match multiple supplying peers with a requesting peer, (2) How to aggregate and coordinate contributions from peers, (3) How to adapt to peer failures and network conditions, and (4) How to efficiently disseminate contents into the network.

In addition, peers are known to be rational, utility-maximizing, agents whose utilities are typically not aligned with the system objectives. Therefore, peers rationality should be considered in designing the protocols. Furthermore, large P2P environments include many unknown-to-each-other peers, which raises the problem of peers fidelity and trustworthiness. Our work also addresses the problem of dynamically assessing peers trustworthiness and incorporating this trust into the protocols. In the following we summarize our approach for attacking these research problems in the context of the two settings mentioned above: the cooperative and the infrastructure environments. (The related work is removed due to space limitations.)

MM'03, November 2–8, 2003, Berkeley, California, USA..
ACM 1-58113-722-2/03/0011.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—Distributed Applications

General Terms

Design, Measurements, Performance, Security, Economics

Keywords

Peer-to-Peer Systems, Multimedia Streaming, Economics of Peer-to-Peer Systems, Distributed Trust Management

2. RESEARCH DIRECTIONS

2.1 PROMISE: Pure P2P Architecture

This section presents our solution to the problem of media streaming in a *cooperative* peer-to-peer environment. The solution is purely distributed. In [5], we proposed PROMISE: A PeeR-tO-peer MedIa StrEaming system. PROMISE architecture consists of a set of peers (end-system machines) interconnected through a P2P substrate such as Pastry, Chord, or CAN. The P2P substrate maintains connectivity among live peers, manages peers joining and leaving, and performs object lookup function. PROMISE assumes that peers exhibit quite heterogeneous characteristics and they do not offer server-like functionalities: they fail more often, offer limited capacity, and may reduce their sending rates unexpectedly. Therefore, several supplying peers may be needed to serve a requesting peer. The design of PROMISE relies on a new application level service called *CollectCast*. CollectCast carefully chooses the supplying peers and orchestrates them in order to yield the best quality for the receiver. Specifically, The major functions of CollectCast are: (1) it infers and leverages the underlying network topology and performance information for the selection of senders. This is based on a novel application of several network tomography techniques; (2) it selects the "best" supplying peers that will likely yield the best quality for the streaming session; (3) it adaptively assigns rate and data to suppliers based on the capacity of each supplier and the current network conditions; and (4) it monitors the status of peers and connections and transparently switches suppliers so that the full quality is maintained throughout the session.

We evaluate PROMISE through simulation and Internet experiments [5]. We show that streaming from multiple failure-prone peers in a dynamic network environment is indeed feasible. Specifically, we show that the full quality can be maintained in presence of failures and packet losses. Our simulation demonstrates that a significant gain in quality (quantified by aggregate received rate and its stability) can be achieved using our tomography-based peer selection technique. Finally, PROMISE does not burden the participating peers: we show that, on the average, a sending peer contributes less than a quarter of the required streaming rate.

2.2 Hybrid Architecture

The hybrid architecture can be used as infrastructure for on-demand media streaming in environments such as distance learning (e.g., distributing lectures to students at several campuses) [3, 4]. The key ideas in this work are: (1) *Network-aware* peers organization, in which a two-level clustering technique is used. In the first level, nearby peers are grouped into a network-cluster. Then, in the second level, network clusters within an autonomous system (AS) are assigned to the same AS-cluster. This peers organization is validated by statistics collected and analyzed from real Internet data. (2) Leveraging peers clustering to implement network-aware searching (to locate nearby suppliers) and dispersion algorithms. The dispersion algorithms efficiently disseminate the files into the system with the objective of keeping sufficient copies of the media files as close as possible to the clients (within the cluster). Network-aware searching and dispersion yield two desirable effects: (i) reduction of the load on the underlying network, since the traffic traverses fewer number of hops, and (ii) better streaming service because the delay is shorter and less variable. (3) Making good use of peers heterogeneity: more powerful peers (super peers) do more work, they help in the searching and dispersion functions. This is why we call this architecture as a hybrid architecture.

We demonstrate the potential of the proposed architecture for a large-scale on-demand media streaming service through an extensive simulation study on large, Internet-like, topologies [3]. Starting with a limited streaming capacity (hence, low cost), the simulation shows that the capacity is rapidly increased and many clients can be served even if they come according to different arrival patterns such as constant rate arrivals, flash crowd arrivals, and Poisson arrivals. Furthermore, the simulation shows that a reasonable client-side initial buffering of 10-20 seconds is sufficient to ensure the full-quality of playback in presence of supplying peers failure and degradation.

An approximate cost-profit analysis to show the economic potential of a large-scale media streaming service built on top of a P2P infrastructure is given in [4].

2.3 Incentive-Compatible Provisioning

In [2], we identify the key economic-related research problems that need to be addressed to realize the great potential of collaborative P2P systems. One of the key problems is the P2P *network provisioning*. In the P2P context, we define network provisioning as the process of creating and distributing information resources among supplier peers in preparation for client requests. This definition bears some similarities with the replication problem addressed by Choen and Shenker [1]. The authors propose optimal replication strategies that minimizes the expected search size in an unstructured P2P environment. They prove that replicating objects in proportion to the square-root of their query rates yields the minimum expected search size for locatable items. However, the replication strategies assume "full cooperation" from peers, in the sense that a peer voluntarily commits some of its capacity to the system and follows the prescribed protocol for replicating objects. However, nodes in P2P systems are found to be economically *rational* or utility-maximizing. Unless properly incentivized, nodes may deviate from the protocol or not participate at all. In the replication problem, if peers are not paid for sharing data, they may not have an incentive to cache data for subsequent sharing. If they do get paid, their decision to cache may depend on more subtle factors such as their "forecast of demand" for the cached materials. In a multi-product environment, a peer may prefer to only cache what it perceives to be the most popular and therefore the most lucrative content, even though this may not be in line with the overall system objective. Peers rationality makes the provisioning problem more challenging. To address this challenge, we are working on the following research tasks: (1) Developing models to understand peers behavior and more specifically their valuation of their own capacity, (2) Studying how the system performance depends not only on the aggregate capacity but also on the heterogenous contributions from individual peers, (3) Analyze the interaction between the provisioning algorithms and the searching algorithms, and (4) Design an incentive mechanism to integrate the above issues into a provisioning algorithm that optimizes a system-wide objective function. Our solution will be based on ideas from microeconomic theory, especially on the theory of mechanism design and its applications in computer science problems.

2.4 Trustworthy P2P Systems

Security plays a critical role on the growth of a collaborative P2P system. More users become an integral part of the system when they are certain that the shared data is not corrupted and they can trust other peers for possible content sharing and exchange. Trust is of particular importance in P2P environments full of unknown entities while lack of predefined trusted relationship (e.g., trusted third parties). It is important for the supplier and the customer to establish a certain level of trust between them before the transaction is conducted. We aim to dynamically assess the trustworthiness of peers and incorporate that into the searching and dispersion algorithms. Three main research tasks are proposed: (1) Evidence identification. What may serve as an evidence in P2P systems? As an example, the fraction of time a peer actually fulfilled its commitment for serving a file or paying the price for consuming a file can be considered as an evidence, (2) Evidence collection. After identifying an evidence, how can we collect sufficient instances of this evidence to: (3) Build trust models. How to use evidences from multiple peers to construct the *web of trust* among them. And finally, how to integrate the web of trust with the searching and dispersion algorithms.

3. REFERENCES

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

[2] M. Hefeeda, P. Afeche, and B. Bhargava. Economics of a collaborative peer-to-peer infrastructure for content distribution. Technical report, CS-TR 03-015, Purdue University, April 2003.

[3] M. Hefeeda, B. Bhargava, and D. Yau. A hybrid architecture for cost-effective on-demand media streaming. *Revised for Journal of Computer Networks*, June 2003.

[4] M. Hefeeda, A. Habib, and B. Bhargava. Cost-profit analysis of a peer-to-peer media streaming architecture. Technical report, CERIAS TR 2002-37, Purdue University, June 2003.

[5] M. Hefeeda, A. Habib, B. Boyan, D. Xu, and B. Bhargava. PROMISE: peer-to-peer media streaming using CollectCast. In *ACM Multimedia 2003 Conference*. To appear.

[6] D. Xu, M. Hefeeda, S. Hambrusch, and B. Bhargava. On peer-to-peer media streaming. In *Proc. of IEEE ICDCS'02*, Vienna, Austria, July 2002.