Multimedia Systems 9: 170–184 (2003)
Digital Object Identifier (DOI) 10.1007/s00530-003-0088-1

Multimedia Systems

Measuring and analyzing the characteristics of Napster and Gnutella hosts

Stefan Saroiu, Krishna P. Gummadi, Steven D. Gribble

Department of Computer Science and Engineering, University of Washington, Seattle, WA, 98195-2350, USA

Abstract. The popularity of peer-to-peer multimedia file sharing applications such as Gnutella and Napster has created a flurry of recent research activity into peer-to-peer architectures. We believe that the proper evaluation of a peer-to-peer system must take into account the characteristics of the peers that choose to participate in it. Surprisingly, however, few of the peer-to-peer architectures currently being developed are evaluated with respect to such considerations. In this paper, we remedy this situation by performing a detailed measurement study of the two popular peer-to-peer file sharing systems, namely Napster and Gnutella. In particular, our measurement study seeks to characterize the population of end-user hosts that participate in these two systems. This characterization includes the bottleneck bandwidths between these hosts and the Internet at large, IP-level latencies to send packets to these hosts, how often hosts connect and disconnect from the system, how many files hosts share and download, the degree of cooperation between the hosts, and several correlations between these characteristics. Our measurements show that there is significant heterogeneity and lack of cooperation across peers participating in these systems.

1 Introduction

The popularity of peer-to-peer file sharing applications such as Gnutella and Napster has created a flurry of recent research activity into peer-to-peer architectures [1–6]. Although the exact definition of "peer-to-peer" is debatable, these systems typically lack dedicated, centralized infrastructure, but rather depend on the voluntary participation of peers to contribute the resources from which the infrastructure is constructed. Membership in a peer-to-peer system is ad-hoc and dynamic: as such, the challenge of such systems is to figure out a mechanism and architecture for organizing the peers in a way that they can cooperate to provide a useful service to the community of users. For example, in a file sharing application, one challenge is organizing peers into a cooperative, global index so that all content can be quickly and efficiently located by any peer in the system [2–4, 6].

To evaluate a proposed peer-to-peer system, the characteristics of the peers that choose to participate in the system must be understood and taken into account. For example, if some peers in a file-sharing system have low-bandwidth, high-latency network connections to the Internet, the system must be careful to avoid delegating large or popular portions of the distributed index to those peers, for fear of overwhelming them and making that portion of the index unavailable to other peers. Similarly, the typical duration that peers choose to remain connected to the infrastructure has implications for the degree of redundancy necessary to keep data or index metadata highly available. In short, the system must take into account the suitability of a given peer for a specific task before explicitly or implicitly delegating that task to the peer.

Surprisingly, however, few of the architectures currently being developed are evaluated with respect to such considerations. We believe that this is, in part, due to a lack of information about the characteristics of hosts that choose to participate in peer-to-peer systems. We are aware of a single previous study [7] that measures only one such characteristic, namely the number of files peers share. In this paper, we remedy this situation by performing a detailed measurement study of the two most popular peer-to-peer file sharing systems, namely Napster and Gnutella. The hosts that choose to participate in these systems are typically end-users' home or office machines, located at the "edge" of the Internet.

Our measurement study seeks to precisely characterize the population of end-user hosts that participate in these two systems. This characterization includes the bottleneck bandwidths between these hosts and the Internet at large, IP-level latencies to send packets to these hosts, how often hosts connect and disconnect from the system, how many files hosts share and download, and correlations between these characteristics. Our measurements consist of detailed traces of these two systems gathered over long periods of time – four days for Napster and eight days for Gnutella, respectively.

There are two main lessons to be learned from our measurement results. First, there is a significant amount of heterogeneity in both Gnutella and Napster; bandwidth, latency, availability, and the degree of sharing vary between three and five orders of magnitude across the peers in the system. This

implies that any similar peer-to-peer system must be very careful about delegating responsibilities across peers. Secondly, peers tend to deliberately misreport information if there is an incentive to do so. Because effective delegation of responsibility depends on accurate information, this implies that future systems must have built-in incentives for peers to tell the truth, or systems must be able to directly measure or verify reported information.

The rest of the paper is structured as follows. Section 2 deals with our measurement technology. Section 2.1 describes the architectures of Napster and Gnutella. Section 2.2 presents the techniques used to crawl these systems. Section 2.3 discusses the active measurement tools used to probe the characteristics of the peers discovered. Our measurement results are described in Sect. 3. Section 4 contains a brief discussion of our results and several recommendations for future file sharing peer-to-peer system designs. Finally, our conclusions are presented in Sect. 5.

2 Measurement methodology

To collect our measurements of Napster and Gnutella, we periodically *crawled* each system in order to gather snapshots the systems' populations. The information collected in these snapshots includes the IP address and port number of each peer in the population, as well as some information about the peers as reported by their software. In this section on the paper, we describe the architectures of Napster and Gnutella, the techniques used to crawl these systems and the tools we used to actively probe the characteristics of the peers. We also reflect on the limitations of our methodology.

2.1 The Napster and Gnutella architectures

Both Napster and Gnutella have similar goals: to facilitate the location and exchange of files (typically images, audio, or video) among a large group of independent users connected through the Internet. In these systems, files are stored on the computers of the individual users (or *peers*), and exchanged using an HTTP-style protocol over a direct connection between the downloading and uploading peers. All peers in these systems are symmetric: they all have the ability to function both as a client and a server. This symmetry distinguishes peer-to-peer systems from many conventional distributed system architectures. Although the process of exchanging files is similar in both systems, Napster and Gnutella differ substantially in how peers locate files (Fig. 1).

In Napster, a large cluster of dedicated central servers maintains an index of the files that are currently being shared by active peers. Each peer maintains a persistent connection to one of the central servers, through which the file location queries are sent. The servers then co-operate to process the query and return a list of matching files and their locations to the peer. After receiving the results, the peer may then select one or more files and locations from this list and initiate file exchanges directly from other peers. In addition to maintaining an index of shared files, the centralized servers also monitor the state of each peer in the system, keeping track of metadata such as the peers' reported connection bandwidth and the duration that the peer has remained connected to the system. This metadata is returned with the results of a query, so that the initiating peer has some information to distinguish possible download sites.

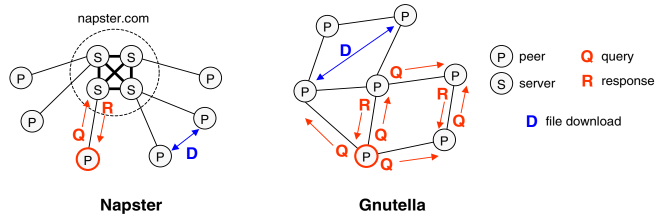

Fig. 1. Locating files in Napster and Gnutella. In Napster, each peer directly queries a central server to discover the location of files. In Gnutella, peers form an overlay network over which file queries are broadcast. In both systems, once a peer discovers the location of a file, that peer downloads the file using a direct TCP connection to the peer that hosts the file

There are no centralized servers in Gnutella, however. Instead, Gnutella peers form an *overlay network* by forging point-to-point connections with a set of neighbors. To locate a file, a peer initiates a controlled flood of the network by sending a query packet to all of its neighbors. Upon receiving a query packet, a peer checks if any locally stored files match the query. If so, the peer sends a query response packet back towards the query originator through the overlay. Whether or not a file match is found, the peer continues to flood the query through the overlay.

To help maintain the overlay as the users enter and leave the system, the Gnutella protocol includes *ping* and *pong* messages that help peers to discover other nodes. Pings and pongs behave similarly to query/query-response packets: any peer that sees a ping message sends a pong back towards the originator, and forwards the ping onwards to its own set of neighbors. Ping and query packets thus flood through the network; the scope of flooding is controlled with a time-to-live (TTL) field that is decremented on each hop. Peers occasionally forge new neighbor connections with other peers discovered through the ping/pong mechanism. Note that it is possible to have several disjoint Gnutella overlays of Gnutella simultaneously coexisting in the Internet; this contrasts with Napster, in which peers are always connected to the same cluster of central servers.

2.2 Crawling the peer-to-peer systems

In this section, we describe the design and implementation of our Napster and Gnutella crawlers. The goal of these crawlers is to produce "snapshots" of these systems by collecting large sets of participating peers.

2.2.1 The Napster crawler

As we did not have direct access to indexes maintained by the central Napster servers, the only way we could discover the set of peers participating in the system at any given time was by issuing queries for files, and keeping a list of peers referenced in the queries' responses. To discover the largest possible set