A Survey of Peer-to-Peer Network Security Issues

James Li

Abstract

In recent years, peer-to-peer (P2P) networks have soared in popularity in the form of file sharing applications. With this popularity comes security implications and vulnerabilities. In this paper, we examining the framework on which most P2P networks are built, and from this, we examine how attacks on P2P networks leverage the very essence of the networks itself: decentralization of resources and of control. Additionally, we look at the privacy and usage attacks that arise in P2P networks as well as approaches that can be used address some of these issues.

Table of Contents

---

1. Introduction

In a traditional computer network, one or more central servers typically provide all of the services available on the network. An example of this is the numerous FTP (File Transfer Protocol) and HTTP (HyperText Transfer Protocol) servers on the Internet that provide file resources for download from clients seeking these services. In contrast to this client-server model of a network, another approach is to distribute the brunt of providing services among the nodes, or peers, such that each node is both a client and a server. This type of network is called a peer-to-peer (P2P) network.

1.1 Definition of P2P

More technically, a P2P network is a special type of computer network that exhibits self-organization, symmetric communication, and distributed control [Risson04]. The network is self-organizing in that there is typically no centralization of resources. As a result, link capacity is typically distributed throughout peers in the network, and as a result control is distributed, as well. As such, the P2P network model stands in direct contrast to the traditional client-server networking model. Whereas a client-server network requires that the server has copious link capacity to feed clients, a P2P network pools the resources of each peer for the common good. However, due to the decentralized and peer-relying nature of P2P networks, they are also susceptible to attacks, which we will explore in this paper.

First, we present some background on P2P networks, including its inception, rise in popularity as an application, and its

querying structure. Next, we examine different ways that P2P networks are often attacked, including denying services, contaminating the network, and compromising personal information of the peers. Finally, we look at some solutions to the attacks and security issues.

Back to Table of Contents

---

2. Background of P2P Networks

The notion of P2P was first established in 1969, in the first Request for Comments, RFC 1. The RFC implies a "host-to-host" connection, indiscriminate of a client-server categorization, which provides responses in the fashion of teletype (TTY) terminals [Peer07] [Crocker69]. However, the first true implementation of a P2P network was Usenet, developed in 1979 [Sundsted01]. In Usenet, while end-user clients still access resources through servers, servers themselves peer with each other in the fashion of a P2P network, sending messages to each other on demand without a central authority [Usenet07].

2.1 Applications of P2P Networks

Since the late 1990s, there has been a surge of popularity in P2P network applications, mainly in the form of file sharing applications used to exchange multimedia files. Some of the most popular and high-profile file sharing protocols include Freenet, Napster, Direct Connect, Gnutella, eDonkey2000, and BitTorrent. By some estimates, file sharing accounts for more traffic than any other application on the Internet [Kurose05]. By far, the recent rise in research interest generated in the P2P field has come from the popularity of file sharing systems. Below is a table of the timeline of development of the most influential of these P2P protocols [Peer07]:

First released	P2P Protocol
July 1999	Freenet
September 1999	Napster
November 1999	Direct Connect
March 2000	Gnutella
September 2000	eDonkey2000
April 2001	BitTorrent

Table 1: timeline of first release dates of popular P2P protocols

Interestingly, the early file sharing application, Napster, was really more of a directory service than a pure P2P system. Nonetheless, Napster opened the way to more advanced approaches to file sharing, as seen with the subsequent of applications such as Gnutella, eDonkey, and, currently the most popular, BitTorrent. While these applications are all considered P2P applications, peer and resource discovery is a distinguishing feature of different P2P networks, as explained below.

2.2 Centralized Directory

One major issue with any P2P system is the discovery of peers and resources in the network. Since there are no fixed servers, peers must rely on some method to locate fellow peers. The most basic approach is a centralized directory where resources are indexed on a central server, and peers query this server for a lookup to find the peer with the desired resource, then make a connection to the peer [Kurose05]. This approach was taken by Napster, for example. BitTorrent also uses a centralized directory server, calling it the tracker. Note that while resource lookup is still client-server, the actual resource transmission, which accounts for the bulk of the network capacity usage, is still P2P. Below is a diagram of the basic layout of this type of network:

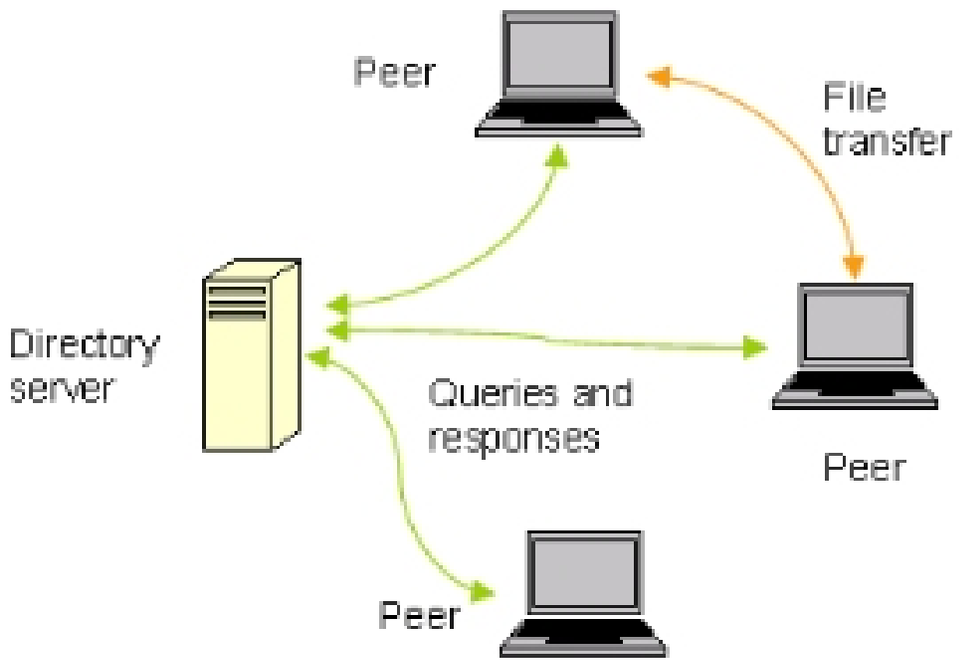

Figure 2: a centralized directory server network

2.3 Query Flooding

Another approach towards peer discovery is query flooding, which is used by newer applications such as Gnutella. The premise here is that instead of relying on a central directory server, a peer would directly broadcast a query to the network, and whomever has the desired resource would respond. Notably, in this approach, there is no central point of failure. However, flooding the network has bandwidth usage considerations that could as well lead to an unintended self distributed denial-of-service (DDoS) attack on the network (a network storm). A variant of the query flooding approach is to select certain, high-availability and high-capacity nodes, as supernodes. These supernodes are given the task of indexing peers within its own domain and answering and creating queries from and to other supernodes. This approach reduces bandwidth usages by a large margin, but it does not really remove the inherent problems with query flooding [Kurose05].

2.4 Distributed Hash Table

Distributed hash tables (DHT) have been introduced around 2001 via the projects Chord, Kademlia, Pastry, and Tapestry. A DHT is essentially a hash table, possessing key-value lookup functionality, with the index distributed among peers in a group. There are variations in the hash function, but the general idea is to minimize the number of peer lookups upon querying for a resource. Typically, *O(log n)* lookups given *n* nodes are needed for a query [Bala03]. DHT systems essentially distribute the centralized directory approach, eliminating a single point of failure. Most of the newer P2P protocols, including trackerless BitTorrent, have been updated to support DHT lookups. Below, we have a sample lookup operation originating from node 2 until the desired resource is found on node 37:

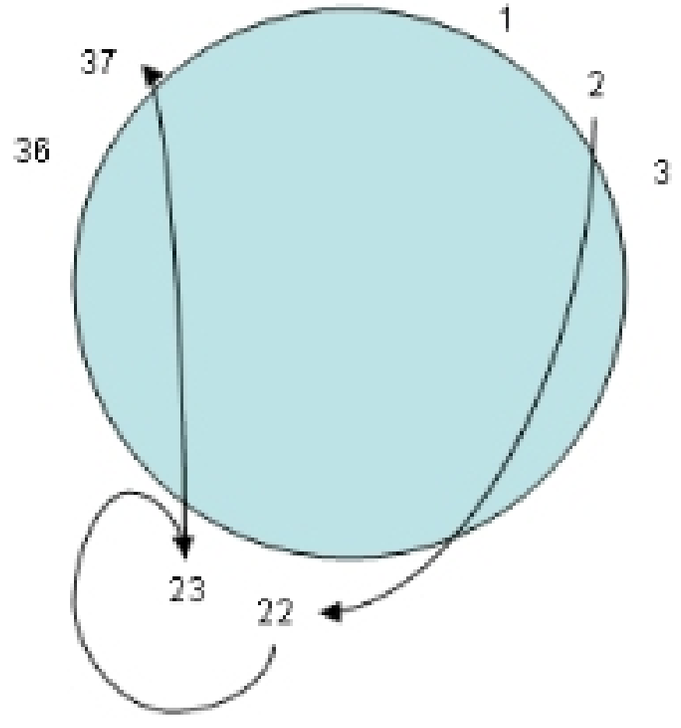

Figure 2: a DHT query from node 2 to 22 to 23 to 73.

Despite these different methods of querying, the actual transmission of resources is still done in P2P fashion, whether using centralized directory, query flooding, or DHT. Unfortunately, attacks have been found that can be used to disrupt or disable the P2P network.