

CHAPTER 02

THE PEER-TO-PEER TECHNOLOGY

1. Introduction :

This chapter defines and introduces the domain of Peer-to-Peer (P2P) networking. It gives a short overview of the evolution of different P2P systems and how resource discovery can be implemented. At the end of this chapter, it summarises some advantages and disadvantages of P2P compared to traditional client/server architecture.

2. What is P2P :

A peer-to-peer (or P2P) computer network is a network that relies on computing power at the edges (ends) of a connection rather than in the network itself [6].

The P2P concept is currently sweeping through both the computing industry and the media, and the implementation is commonly seen in instant messaging (IM) applications such as ICQ or MSN Messenger, and different file sharing applications, such as Kazaa or Gnutella. Conceptually, P2P is much more – or much less – than that, P2P can simply be two or more PCs that are connected and share resources without going through a separate server. At the other extreme, one could say that the entire Internet operates very similar to a giant P2P network. From a broad perspective, the whole Internet itself consists of networked computers containing a wide selection of geographically separated data, communicating directly with one other. [7]

P2P could be explained by answering the question: “How can you connect a set of devices in such a way that they can share information, resources, and services?”[8]. It seems like an easy question, but if we dig deeper, we learn that this basic question derives more complex challenges:

How does one device learn from another device’s presence, that is, how do we deal with discovery?

- How do devices organize groups of common interest?
- How does a device advertise its resources?
- How do we uniquely identify a device?
- How do devices exchange data?

Many P2P solutions have been created to provide answers to these questions. The problem is that they all have their own answer hard-coded into the implementation, giving no room for flexibility and interoperability. To

evolve P2P into a mature solution platform, developers need to agree on a solid, well-defined base language to communicate and perform the fundamentals of P2P networking.

3. Client/Server Architecture :

In the traditional client/server architecture the client sends a request to the server, which handles most of the processing involved in delivering the requested service, leaving the clients relatively unburdened.

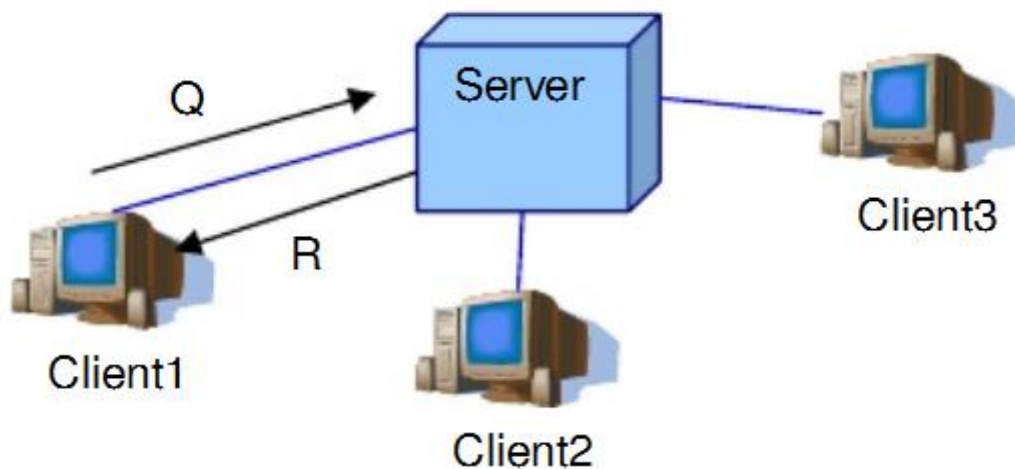


Figure 1 - Client/Server Architecture

Client:

- Sends query request (Q)
- Receive response (R)

Server:

- Receive query request (Q)
- Processes service requests
- Sends the result as a response to the client (R)

This architecture has major drawbacks. As the number of clients increase, the load on the server also increases, until the bandwidth or the processing power reaches its limits, preventing the server from handling additional clients. The advantage however, is that the client is left with very little responsibility, thus it does not require high computing power. Ultimately, this means that almost any device with a network connection can act as a client and receive server data.

4. P2P Network Architectures :

There are mainly three network models of P2P, namely Centralized, Decentralized and Hybrid models.

4.1. Centralized Architecture :

The first generation Peer-to-Peer system was initiated by the launch of Napster in May 1999. When this infamous application was at its peak in February 2001, it had 29.4 million registered users who shared 2.79 billion files in the same month [6]. Napster was based on a centralized index, which ultimately led to its downfall in late 2001, when it was forced by the record industry to shut down.

Centralized network architecture uses a centralized indexed server to maintain a data base of all the content and users at any time. The database is updated whenever a peer logs on to the network.

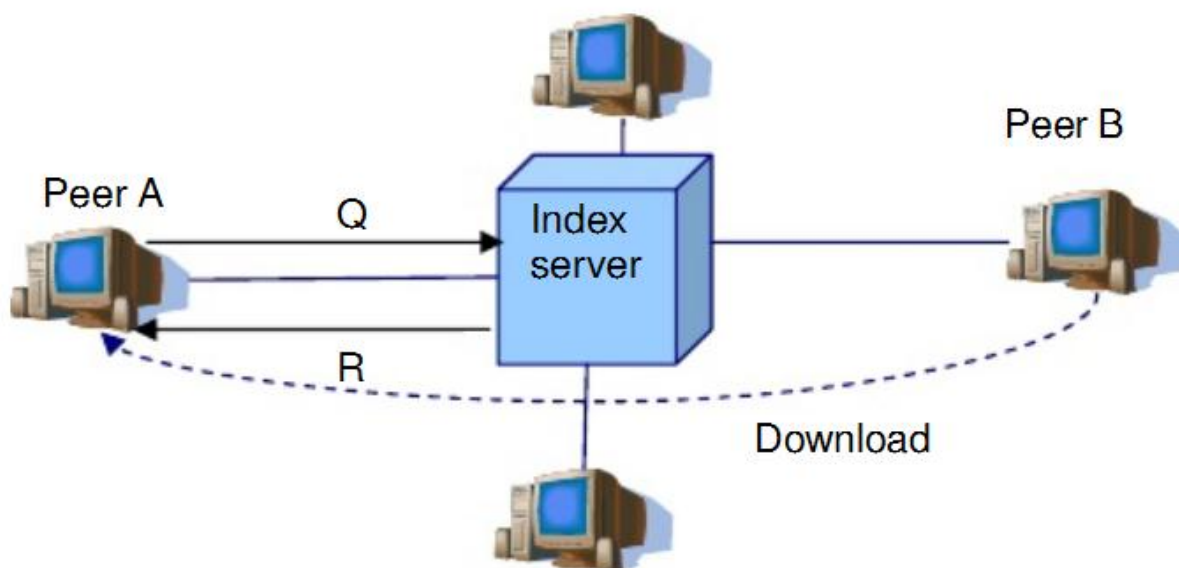


Figure 2 - Centralized Network Architecture

Peer A sends a query request to its index server. The index server uses the search request to query the database. If matches are found, the server returns the result to node A, telling him which node has the file, in this example, Peer B. Node A uses this information to start download from Node B.

Evaluation:

This architecture allows a fast search response time, and is easy to implement and maintain. It provides a high degree of performance and resilience, but has a single point of failure. Because of this, it is vulnerable to censorship and technical failure. Popular data may become less accessible because of the load of the requests on a central server. Another disadvantage is that its central index might be obsolete, because the database is only refreshed periodically.

4.2. Decentralized Architecture :

Second generation P2P uses a decentralized, distributed architecture to avoid the centralized weakness, “single-point-of-failure”. Instead of central servers, each peer acts as an index server, searches and holds its own local resources, and as a router, relaying queries between peers. An example is the Gnutella network.

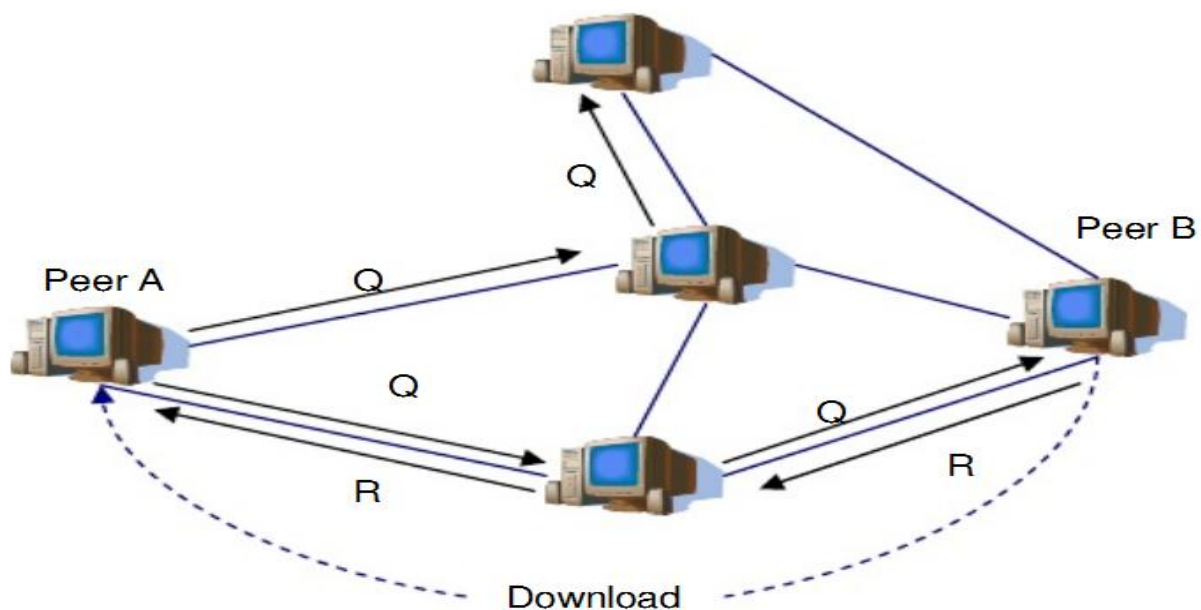


Figure 3 - Decentralized Architecture

Node A sends a query message to the peers it is directly connected to. These peers check their local list of resources to match the query and forward the query to the peers they are connected to on the network. This process continues, spreading the query across the network. If a peer, in this example Peer B, matches the query with its local resource, it returns a response message back across the network to Peer A. Peer A then downloads the resource directly from Peer B.

Evaluation:

Each peer is directly connected to a number of other peers. Relaying queries and result messages between peers generates large network traffic (chatter). It also results in slow information discovery compared to a centralized architecture. The system avoids single point of failure, which means it is resistant to crashing and shutdowns. It also scales inherently.

4.3. Hybrid Architecture :

Third generation P2P is a hybrid of centralized and distributed, combining the best of both architectures. It deploys a hierarchical structure by establishing a backbone network of Super Nodes that take on the characteristics of a central index server. When a client logs on to the network, it makes a direct connection to a single Super Node which gathers and stores information about peer and content available for sharing. An example of a hybrid P2P network is the Direct Connect (DC) network.

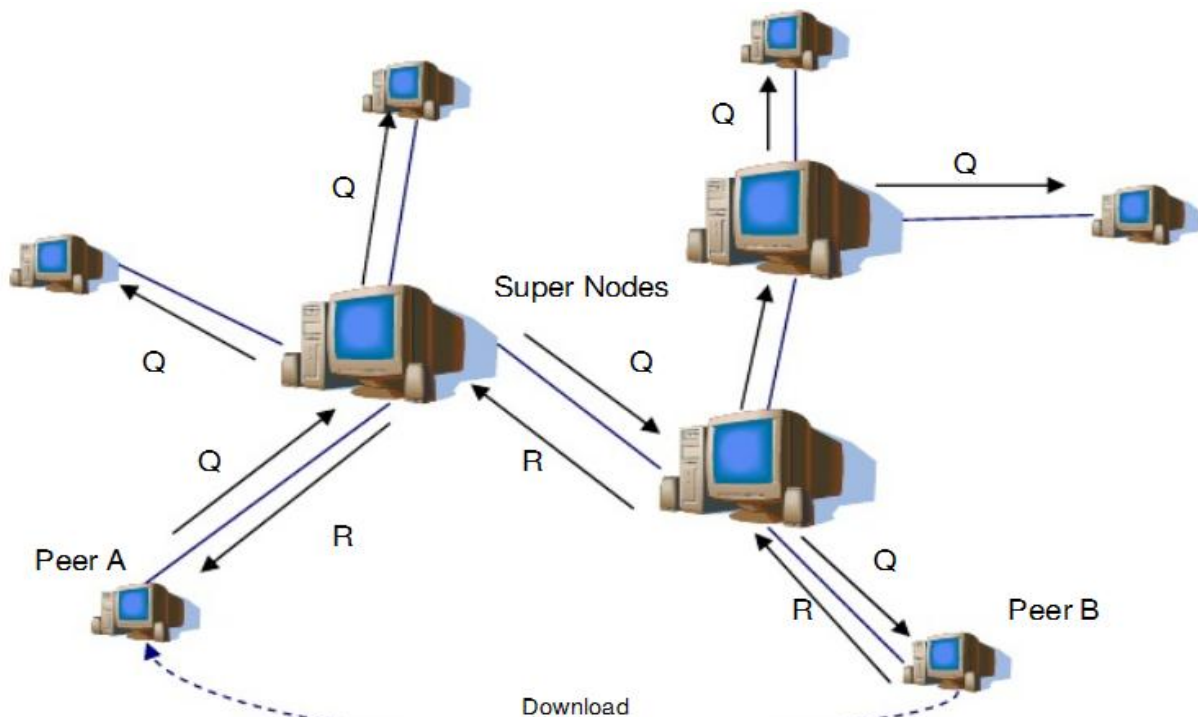


Figure 4 - Hybrid Architecture

Peer A sends a query message to its local Super Node. The Super Node runs the query in its own index and disseminates the query to other Super Nodes on the network. The query response is returned to Super Node which in turn relays the results to Peer A. Peer A then downloads the resource directly from Peer B.

Evaluation:

The use of Super Nodes improves the search response times and generates less overhead traffic on the network than decentralized networks. The Super Nodes also reduce the workload on central servers in comparison with fully centralized indexing systems such as Napster. The single point of failure or control diminishes as the number of Super Nodes increases.

5. Resource Discovery :

Discovering resources can be handled in several ways. Brendon Wilson [8] implies three main methods: No discovery, direct discovery and indirect discovery.

5.1. No discovery :

Peers relay on a cache of previously discovered advertisements. This reduces network traffic, but the information can become obsolete and increase network traffic by trying to discover a resource that no longer exists at given peer and then resort to active discovery. To reduce the possibility of a given advertisement becoming obsolete, a cache can make advertisements expire, there by removing them from the cache based on the probability that a given advertisement is still valid.

5.2. Direct Discovery :

Peers that exist on the same LAN might be capable of discovering each other directly without relying on an intermediate rendezvous peer to aid the discovery process. Direct discovery requires peers to use the broadcast or multicasting capabilities of their native network transport. Unfortunately, this discovery technique is limited to peers located on the same local LAN segment and usually can't be used to discover peers outside the local network. Discovering peers and advertisements outside the private network requires indirect discovery conducted via a rendezvous peer.

5.3. Indirect Discovery :

Indirect discovery requires the use of a Super Peer to act as a source of known peers and advertisements, and to perform discovery on a peer's behalf.

This technique can be used by peers on a local LAN to find other peers without using broadcast or multicast capabilities, or by peers in a private internal network to find peers outside the internal network.

6. Why Peer-to-Peer networking :

The potential of P2P reaches far beyond the recent years media focus on the area, namely through distribution of copyrighted material such as mp3's and movies. The advantages and disadvantages of P2P are usually compared to the traditional client/server technology. Some advantages are:

6.1. Distributed computing power :

In his book, Brendon Wilson [8] presents this example to show the enormous amount of potential computing power and storage we have on client machines around the globe:

“Assume, with a lot of modesty, that 10 million 100 MHz machines are connected to the Internet at any time, each possessing only 100MB of unused storage space, 1000bps of unused bandwidth, and 10% unused processing power. At any time, these clients represent 10 petabytes (10^{15} bytes) of available storage space, 10 billion bps of available bandwidth, and 10^5 GHz of wasted processing power! P2P is the key to realizing this potential” [8].

6.2. No single point of failure:

Removing the centralized server, which can be subject to crash, failure or overload would provide a more robust system which could withstand major disasters or other events that would result in downtime.

6.3. Distributed search:

The Internet is a network of underutilized resources, partly due to the traditional client-server computing model. Take web searching for instance; no single search engine can locate and catalogue the ever-increasing amount of information on the Web at an acceptable speed [9]. Google claims that it searches over 8 billion web pages (June 2005), but this is just a small part of the World Wide Web. Consider all the private storage on client machines, and all

the databases containing data the web engines have no access to. Using P2P technology and giving each Peer a responsibility to search its own domain would produce a much larger, more accurate and more updated search result.

7. Popular Peer-to-Peer Protocols :

7.1. Napster :

Napster was the program which brought peer-to-peer networking to the masses. The original Napster, released in 1999, was a centralized peer-to-peer system which allowed users to share music files on their hard drives, and download files from the other users logged on to the service.

Napster does not exist in its original form anymore because of legal actions against the company. Nowadays, Napster is a subscription based, legal music service, where users can download songs by paying a monthly fee. The file-transfer paradigm used by the current Napster is no more peer-to-peer; it is traditional client-server instead. The original Napster architecture is described in [10], and compared against the Gnutella architecture.

In the original architecture, a centralized database was used to store information about which files every node had, and this database was used for file searches. When a user found an interesting file, the centralized server passed information about the peer having that file, so that peers could directly transfer that song. The idea behind the Napster was intelligent, the company did not have to deal with the huge traffic loads generated by the actual file transfer sessions; it only needed to handle the signaling traffic coming from and going to the centralized database.

7.2. Gnutella :

Gnutella system is a hybrid peer-to-peer content sharing system, designed originally by Nullsoft. Unlike Napster, Gnutella allows users to share any kinds of files, not just music. The Gnutella system tries to fight legal threats by not having centralized servers which can be shut down.

The original Gnutella protocol was a pure peer-to-peer system without any central nodes [10]. Due to scalability issues, the concept of ultra-peers was introduced in the Gnutella version 0.6 [11]. Some nodes in the Gnutella network are assigned as ultra-peers, this assignment is based on the node resources: the network bandwidth, the firewall/Network Address Translation (NAT) status and the uptime the node is having. Many end nodes connect to these ultra-peers like

ordinary nodes connected to the Napster servers. The ultra-peers form a pure peer-to-peer network among themselves; thus, they work as proxies to the Gnutella network for the less capable nodes. In the new architecture, the less capable nodes do not need to bother with large amount of signaling traffic, whereas the more capable nodes function as super-peers and are responsible for propagating search messages inside the network.

7.3. Freenet:

Freenet is a pure peer-to-peer system designed by Ian Clarke. Freenet's main aim is to provide anonymity for its users. It allows users to publish and fetch files anonymously in the network. Freenet provides privacy via strong encryption.

Content is distributed over the network, and users of the network are not able to deduce what information is passed via their computers, nor what files the Freenet system has stored on their computers. The system replicates files in the network automatically, so a computer can store files that the user has never requested. Instead of sending simple flooded requests as searches, Freenet builds a dynamic routing table containing mappings between the addresses of other nodes, and the content those nodes are assumed to be holding. However, files are always routed via multiple nodes, so neither the sender nor the receiver of the file knows who has the file or who is requesting it[12].

8. Conclusion:

We presented in this chapter an overview on peer-to-peer. However, it is the decentralized architecture that will be used later in our project.

Adobe AIR, which will be discussed later, will be the base platform that we will adopt in our project.

In Chapter 03, we will try to understand in more detail how peer-to-peer. But also understand the benefit of such technology and the opportunity it offers us in the future.