

Architectures of Network Applications

Gustavo Hammerschmidt*
California State University San Marcos

January 30, 2020

Abstract

This article discourse differences in architectures of diverse networks, the protocols used on the relations between computers and examples of applications that use these architectures.

1 Introduction

We are surrounded by information. Users connect each day to the internet for performing various activities, most of them have never written a line of code or know any programming languages. The internet is such a success that new and old technologies exist and coexist without having to change as the internet enhances itself[3], as the computer hardware gets better. Between users and servers, a lot of protocols are followed in order for a message to be delivered. The most known architectures who support this environment are Client-server and the peer-to-peer architectures, there is also a hybrid architecture composed of both.

2 The Client-Server Architecture

2.1 Description

This architecture was first conceived by the company Xerox PARC during the 70s. Examples of this model are Email and the World Wide Web. Two protocols that use this concept are HTTP(HyperText Transfer Protocol) and FTP(File Transfer Protocol).

*hamme032@cougars.csusm.edu

The client-Server [7][11] Architecture presumes that a client will ask a server for some resource, information or service, therefore, the server is awaiting a client initial message. The servers execute applications and share information with clients, who communicate to gain access to server's resources. The communication between them is held via computer networks; sometimes both client and sever are in the same computer.

2.2 The HyperText Transfer Protocol

HTTP[6] is used on the World Wide Web to transfer data across the network. It is used to send pages or files written in HTML. In order for it to be able to transfer data, the protocols TCP and IP(as known as TCP/IP) need to be already established – this protocol is responsible for the internet communication between servers e users. HTTP is based on the request-response paradigm, which is the client-server architecture.

2.3 The File Transfer Protocol

FTP[10][2] is used on communications between computers and servers to transfer files, it is also based on client-server architecture and is used to transfer different types of files, including videos, documents and music.

3 The Peer-to-Peer Architecture

3.1 Description

The Peer-to-Peer Architecture [8][5] is a network where computers play both client and server roles, which allows them to communicate in a decentralized connection. These devices form a virtual network through their routing mechanism.

The topology(overlay network) of this network influences some of its attributes – such as its performance –; in this topology graph, nodes represent the computers and the edges represent the communication channels. Algorithms are then implemented to find nodes and connections, this ensures that every node will be able to send or receive data to other nodes in the network. There are two types of overlay networks: unstructured and structured.

3.2 The Peer-to-Peer unstructured Architecture

The unstructured topology of a peer-to-peer network is made of algorithms that build a list of close-related devices randomly. When a node needs to find another

node, a breadth-first search is made through the entire network, though some networks have mechanisms to limit the depth of a search: if the searched node is too far, it may not be found. Gossip protocol is an example of data spreading in an unstructured peer-to-peer architecture.

3.2.1 The Gossip Protocol

The gossip protocol [4] is a way of spreading data to all members in a group. The data is passed from one node to its close-related nodes. This way of sharing is also known as epidemic protocol.

3.3 The Peer-to-Peer structured Architecture

The structured topology is build from a deterministic procedure: generally hash tables. In this architecture, each node has a random key that is used to find the request in a hash table and then return the correspondent node, so a connection can be made. A example of this architecture is the BitTorrent's application.

3.3.1 The BitTorrent's application

BitTorrent is an application for file sharing. The user, who wished to use it, needs to use a BitTorrent client(most famous of them is uTorrent) to access the trackers. The trackers provide lists of files and the client can select one peer user to transfer a file.

4 Visual Representation of the Architectures

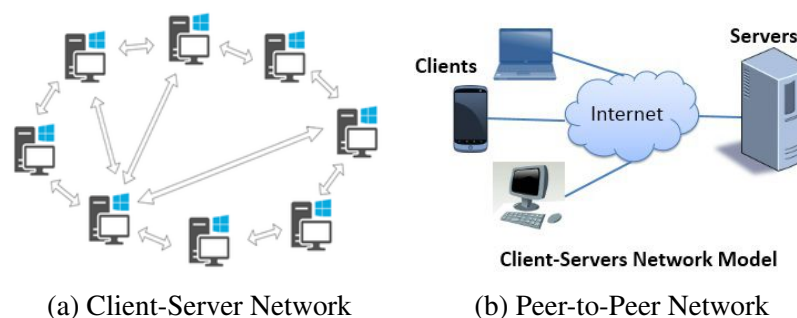


Figure 1: Architecture of the Networks

In figure one, we see the representation of both architectures. The figure demonstrates how devices are connected to other devices(hosts) in a client-server

Peer-to-Peer Networks	Client-Server Networks
Easy to set up	More difficult to set up
Less expensive to install	more expensive to install
Low levels of security	High levels of security

Table 1: Client-Server vs Peer-to-Peer Architecture

network. The figure b demonstrates how devices connect to each other on a Peer-to-Peer Network.

5 Client-Server Architecture vs Peer-to-Peer Architecture

On table 1, some comparisons are made. First topic, a p2p(Peer-to-Peer) network is easily set up when compared to a client-server one, because devices set their selves up when joining the network: all they need is a list of close-related devices; but to a client-server one, there must be a server side of a application set on a host computer running and the clients need to have their part of the application ready to use. This difficulty on the setting-up stage implies in higher costs too: basically in a p2p network there is no cost involved for the application developers in a long-term, once servers are not needed. When it comes to security, it is safer to use a client-server network: the server side of the application is controlled by the company, the access is a lot more restricted, reverse engineering is not as effective in this situation, not only that, a bad-intent user can spread viruses in a peer-to-peer via gossip protocols in an easier way, every node in the network is exposed to these threats.

6 The Hybrid Architecture

6.1 Description

The hybrid architecture [1] is a mix of client-server and peer-to-peer architectures. This architecture is found on applications where the user connect to a server to find another user, such as Skype or face-time calls applications; the user first communicate with a server via a client-server network and then, when it receives an address to another user, it uses peer-to-peer to communicate with him. Spotify [9] used this architecture at the very beginning as a way to spread music across users,

currently this architecture has been replaced by a client-server one, because the company has enough servers around the world now.

7 Conclusion

Now that both architectures have been explained, which one to adopt in a next project? Truth must be told that there is not a right and wrong approach in this story. There are cases that would perform better in one architecture instead of another. Positive and Negative aspects of both should be considerate before selecting one of them. In my opinion, security in network environments is critical for having a good application: as in the Spotify initial-stage[9], where the company did not have enough servers around the world, using a hybrid architecture is justifiable for means of company increases in reach-ability and market competitiveness; when applications reach a big dimension and important data from users are capable of being exposed, developers should consider a safer way to transmit messages through networks. Peer-to-Peer networks are also good approaches to small problems that can be resolved without a dedicated server, IOT devices are a good example of this: having a decentralized network of devices that can communicate with each other is the core matter of a Internet of Things – in a peer-to-peer IOT, the cost of such network would be low compared to a client-server one.

8 Table of contents

Contents

1	Introduction	1
2	The Client-Server Architecture	1
2.1	Description	1
2.2	The HyperText Transfer Protocol	2
2.3	The File Transfer Protocol	2
3	The Peer-to-Peer Architecture	2
3.1	Description	2
3.2	The Peer-to-Peer unstructured Architecture	2
3.2.1	The Gossip Protocol	3
3.3	The Peer-to-Peer structured Architecture	3
3.3.1	The BitTorrent's application	3

4	Visual Representation of the Architectures	3
5	Client-Server Architecture vs Peer-to-Peer Architecture	4
6	The Hybrid Architecture	4
6.1	Description	4
7	Conclusion	5
8	Table of contents	5

References

- [1] Chris Carter. “HYBRID CLIENT-SERVER, PEER-TO-PEER FRAMEWORK FOR MMOG”. In: (). DOI: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5583228>.
- [2] Cloudwards. *What Is FTP? Everything About File Transfer Protocol*. 2018. URL: <https://www.cloudwards.net/what-is-ftp/>.
- [3] Douglas E Comer. *Computer Networks and Internets (6th Edition)*. Pearson, 2014.
- [4] Mark Jelasity. “Gossip”. In: (). DOI: <http://publicatio.bibl.u-szeged.hu/1529/1/gossip11.pdf>.
- [5] Manuel Costa Miguel Castro and Antony Rowstron. “Peer-to-peer overlays: structured, unstructured, or both?” In: (). DOI: <https://www.microsoft.com/en-us/research/wp-content/uploads/2004/07/structella-tr.pdf>.
- [6] OficinaDaNet. *The HTTP Protocol*. 2007. URL: https://www.oficinadanet.com.br/artigo/459/o_protocolo_http.
- [7] Haroon Shakirat Oluwatosin. “Client-Server Model”. In: (2014). DOI: <https://pdfs.semanticscholar.org/e1d2/133541a5d22d0ee60ee39a0fece970a4.pdf>.
- [8] Riidiger Schollmeier. “A Definition of Peer-to-Peer Networking for the Classification of Peer-toPeer Architectures and Applications”. In: (). DOI: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=990434&tag=1>.

- [9] TechCrunch. *Spotify removes Peer-To-Peer Technology From Its Desktop Clients*. 2014. URL: <https://techcrunch.com/2014/04/17/spotify-removes-peer-to-peer-technology-from-its-desktop-client/>.
- [10] Technopedia. *File Transfer Protocol (FTP)*. URL: <https://www.techopedia.com/definition/1872/file-transfer-protocol-ftp>.
- [11] Wikipedia. *Client-server model*. URL: https://en.wikipedia.org/wiki/Client%E2%80%93server_model.