

Leontiadis Ilias

1. What is the definition of a p2p system given by the authors in sec 1?

This is the definition given by the authors:

The term “peer-to-peer” refers to a class of systems and applications that employ distributed resources to perform a critical function in a decentralized manner.

The resources encompass computing power, data (storage and content), network bandwidth, and presence (computers, human, and other resources). The critical function can distributed computing, data/content sharing, communication and collaboration, or platform services. Decentralization may apply to algorithms, data, and meta-data, or to all of them.

Compare it with at least one of the definitions surveyed in the last paragraph of pg 2.

“the sharing of computer resources and services by direct exchange between systems”

Compared to the writers definition there is no reference to the critical function that is the goal of the p2p system. Also it is too short and it does not describe the type of the resources that can be exchanged.

“the use of devices on the internet periphery in a nonclient capacity”

This definition is too general. It has nothing in common with the one given by the authors. It is limiting p2p systems only to the internet and it is not saying anything about distribution of resources.

“P2P is a class of applications that takes advantage of resources – storage, cycles, content, human presence – available at the edges of the Internet. Because accessing these decentralized resources means operating in an environment of unstable connectivity and unpredictable IP addresses, P2P nodes must operate outside the DNS system and have significant or total autonomy from central servers”

It includes many of the main points of the author’s definition. It includes the extra information about the dns... It does not state anything about the ‘critical function’

«defines P2P systems as those with independent Lifetimes».

Another too short definition that cannot give any clue to someone about what p2p system really are. Nothing in common with the authors.

2. In Fig 2 (pg 3), the authors compare some aspects of the client-server and the p2p computing models. List and explain these aspects.

managed	self-organized	Peers are autonomous and thus self-organized. In C/S architecture the server manages the connections.
configured	ad-hoc	Client/server system architecture is pre-configured. In p2p systems the whole architecture is dynamic.
lookup	discover	Peers discover resources since they have to search it using algorithms in a distributed way. On the other hand, clients just look it up in the servers index.
hierarchy	mesh	C/S architectures are structured. There is a client and a server → hierarchy. However in p2p there are just meshes (connections) between peers.
static	mobile	Peers can move. Servers cannot.
IP-centric	also non-IP	Client/Server architectures need an IP id to find the server.
DNS-based	custom naming	The paper states that p2p systems use their own naming protocols.
RPC	Async	We can only use asynchronous communication in p2p systems.
server dependencies	independent lifetime	Clients depend from the server.

3 What is a hierarchical and what is a flat client-server model?

The client-server model can be flat where all clients only communicate with a single server (possibly replicated for improved reliability), or it can be hierarchical for improved scalability. In a hierarchical model, the servers of one level are acting as clients to higher level servers. Examples of a flat model include traditional middleware solutions, such as object request brokers and distributed objects. Examples of a hierarchical model include DNS server and mounted file systems.

4 What is a super peer?

SuperPeers contain some of the information that others may not have. Other peers typically lookup information at SuperPeers if they cannot find it otherwise. Example of such solution is Kazaa.

5 What is the difference between a compute-intensive and a componentized application?

In contrast to compute-intensive applications that run the same task on many peers, componentized applications run different components on each peer.

How does this relate to vertical and horizontal distribution?

Horizontal distribution → Compute intensive. Because in horizontal distribution every peer has the same type of data/job but in different data set.

Vertical distribution → componentized applications. Because in vertical distribution we distribute the type of data/jobs to multiple peers and not the data.

6 What is according to the authors the main challenge of communication in p2p?

The fundamental challenge of communication in a P2P community is overcoming the problems associated with the dynamic nature of peers. Either intentionally (e.g., because a user turns off her computer) or unintentionally (e.g., due to a, possibly dial-up, network link failing) peer groups frequently change. Maintaining application-level connectivity in such an environment is one of the biggest challenges facing P2P developers.

7 What is the most common solution to reliability across p2p systems

By using more traditional database techniques such as replication.

8 What are the advantages/disadvantages of the centralized directory, the flooded requests, and the document routing models

- Centralized directory
 - Advantages:
 - low message overhead
 - low latency
 - easy implemented
 - search success guarantee.
 - Constant time to find a resource.
 - Find many resources and use the best
 - Disadvantages:
 - Not so distributed
 - Scalability
 - server depended
 - law problems
 - needs bigger servers in CPU/storage as request increase.
- Flooding requests:
 - Advantages:
 - pure p2p model
 - fully distributed.
 - Disadvantages:
 - Require network bandwidth
 - scalability problems.
 - May not find data that exist (probabilistic guarantee)
- Document routing:
 - Advantages:
 - Fully distributed
 - Efficient with large global communities.
 - Scalability.

- Disadvantages
 - ID must be known
 - Difficult implementation
 - Islanding problem.

9 In the centralized directory approach, after the best peer is located, the file exchange occurs directly between it and the requesting peer. What are the advantages/disadvantages of this?

Advantages:

- Scalability since the server does not need to be involved in file transfers. It is just an index.
- The peers communicate directly thus the peer can be cached for future use.
- The file transfers are distributed and do not depend on a single server.

Disadvantages:

- Security issues (the “best” peer can be hostile).

10 What can be considered as a closure mechanism in Gnutella?

New nodes must know the address of another Gnutella node or use a host list with known IP addresses of other peers. The node joins the network of peers by establishing a connection with at least one peer currently in the network. Then, it can begin discovering other peers and cache their IP addresses locally.

11 What are the factors that affect scalability, give one example for each

- amount of centralized operations that needs to be performed (e.g, synchronization and coordination)
- the amount of state that needs to be maintained (e.g. index in Napster)
- the inherent parallelism an application exhibits (e.g Parallel algorithms need to perform some tasks in a single machine (such as input/output...))
- and the programming model that is used to represent the computation. (Flooding/Global Directory etc..)

12 Given the ad-hoc nature of connectivity in p2p, comment on what type of (message-oriented) communication (i.e., synchronous/asynchronous, transient/persistent) would be more appropriate

Since the system is dynamic and a peer can go offline at any time, we cannot wait till the message is delivered. We EXPECT to have many messages undelivered. So we need asynchronous communication.

For the same reason we cannot store the messages in order to have persistent communication. If a peer is offline, it is just not going to get the message.

To sum up, we should use asynchronous transient communication.

13 pg 17, 1st column, last par "The geographical distribution of the peers help to reduce congestion on both peers and the network" Explain.

When peers are geographically distributed they choose peers that are near to them in order to connect (due to lower latency and higher speeds), so there is less congestion to the peers and to the network since connections are also geographically distributed.

Moreover when we scale the p2p network geographically we also use more physical networks and thus we distribute the communications.

14 What is the goal of caching in p2p?

- Caching reduces the path length required to fetch a file/object and therefore the **number of messages** exchanged between the peers.
- Reducing such transmissions is important because the communication **latency** between the peers is a serious performance bottleneck facing P2P systems
- to **maximize query throughput**
- **balance the workload** in the system.

What are the advantages/disadvantages of caching the reply at all nodes in the return path?

Advantages:

- Many peers cache the data
- We bring the data "closer" to the nodes that ask it (by caching the path)

Disadvantages

- We need more memory
- We may cache data to peers that never used it (or they are never going to).
- Cache consistency algorithm overhead increases (since more nodes have the data in their cache).

Can you think of any alternatives?

Cache it using a probability, cache it only at the node that search it.

15 What does the "power-law distribution of the p2p network" (pg 17) mean?

That when a peer knows some other peers then using few steps a peer can flood the network and find another peer (unknown to him). This is similar to the "small world phenomenon".

16 Compare/relate the definition of distributed systems in sec 5 2 (pg 21) with sec 1 4 of the textbook

The book focus on the fact that the DS appears to its user as a single coherent system. The authors describe the DS as a way to transfer tasks (computation tasks) from one machine to another in order to increase performance.

They both state that it is a collection of standard machines.

17 Why is the fault tolerance problem a greater challenge in collaborative p2p systems than in file sharing p2p systems?

In shared applications, messages often must be delivered reliably to ensure that all peers have the same view of the information. In some cases, message ordering may be important. While many well-known group communication techniques address these challenges in a non-P2P environment, most P2P applications do not require such strict guarantees.