Alkis Georgopoulos, A.M. 39, 12 May 2003

QUESTIONS
(cover sec 1-5)

1. What is the definition of a p2p system given by the authors in sec 1?
Compare it with at least one of the definitions surveyed in the last paragraph of pg 2.

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 be 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. This does not preclude retaining centralization in some parts of the systems and applications if it meets their requirements. Typical P2P systems reside on the edge of the Internet or in ad-hoc networks.

The Intel P2P working group defines it as "the sharing of computer resources and services by direct exchange between systems"

I think the second definition is both simple and comprehensive, while the first one could also describe a distributed system. I also liked the statement - definition the authors give in pg 3:
P2P is a way to leverage vast amounts of computing power, storage, and connectivity from personal computers distributed around the world.

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.

There is no clear border between a client-server and a P2P model. Both models can be built on a spectrum of levels of characteristics (e.g., manageability, configurability), functionality (e.g., lookup versus discovery), organizations (e.g., hierarchy versus mesh), components (e.g., DNS), and protocols (e.g., IP), etc. Furthermore, one model can be built on top of the other or parts of the components can be realized in one or the other model. Finally, both models can execute on different types of platforms (Internet, intranet, etc.) and both can serve as an underlying base for traditional and new applications. Therefore, it should not be a surprise that there is so much confusion about what P2P is and what it is not.

The authors summarize the similarities between client-server and the p2p computing models, explaining why many aspects of p2p are not considered "new".

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 hierarchal 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?

(SuperNodes στην ορολογία του Kazaa, γιατί παύουν πλέον να είναι ομότιμοι).
SuperPeers contain some of the information that others may not have. Other peers typically lookup information at SuperPeers if they cannot find it otherwise.

5. What is the difference between a compute-intensive and a componentized application? How does this relate to vertical and horizontal distribution?

While compute-intensive applications utilize idle machine cycles in different computers to perform the same task, a componentized application runs different components on each peer.

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.

The most common solution to reliability across P2P systems is to take advantage of redundancy. For example, in case of computeintensive applications upon a detection of a failure the task can be restarted on other available machines. Alternatively, the same task can be initially assigned to multiple peers. In file sharing applications, data can be replicated across many peers. Finally, in messaging applications, lost messages can be resent or can be sent along multiple paths simultaneously.

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

Centralized directory: main node can be a bottleneck, but with fast enough servers it's ok. Also some availability concepts.
Flooded requests: efficient in small networks but has scalability problems.
Document routing: Small number of hops for searching, plus only logN peers need to be updated when a peer joins / leaves the network. Suitable for large networks. The main disadvantage is that the hash value (document ID) must be known prior to searching.

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?

A potential disadvantage is that the "best peer" could not always be available, but this can be avoided if the server responds with many "best peer" addresses. The main advantage is that server traffic is kept low, resulting in high scalability.

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

In fully decentralized file systems, such as Freenet and Gnutella, just finding the network becomes difficult. In Gnutella, for example, 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.

Scalability is limited by factors such as the amount of centralized operations (e.g, synchronization and coordination) that needs to be performed (e.g. LogN in centralized directory), the amount of state that needs to be maintained (e.g. neighbours – addresses cache in flooding), the inherent parallelism an application exhibits (e.g. strip partitioning in Search for ExtraTerrestrial Intelligence @ Home), and the programming model that is used to represent the computation.

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 peer availability is not guaranteed, asynchronous communication should be preferred. Also transient communications can outperform persistent communications due to traffic conditions.

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

The main idea in replication is that if a peer can find the searched data in a nearby peer, network traffic (congestion) is reduced and thus peers utilise their bandwidth downloading, not propagating.

14. What is the goal of caching in p2p? What are the advantages/disadvantages of caching the reply at all nodes in the return path? Can you think of any alternatives? Is this possible in Gnutella?

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.

A serious disadvantage is when the data to be cached is quite large (e.g. a movie). Having many 1Gb – size copies is very insufficient, and thus the different copies should be kept to a minimum.
An alternative would be to cache only small and often asked – for data, while maintaining a list of nearby alternatives for big downloads.

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

It measures the number of edges between 2 vertices (peers) in a p2p network.

16. Compare/relate the definition of distributed systems in sec 5.2 (pg 21) with sec 1.4 of the textbook.

I think the key difference is in
"Hence, we consider distributed computing systems to be P2P systems."

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

Fault tolerance is another challenge for collaborative 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. The primary solution employed in P2P applications is to queue messages that have been sent and not delivered (i.e., because a given peer is down or offline). The messages can then be delivered to the offline peer when it comes back online.