2. What Is This *"Peer-to-Peer"* About?

Ralf Steinmetz (Technische Universität Darmstadt)
Klaus Wehrle (Universität Tübingen)

Currently, a new and highly interesting paradigm for communication on the Internet, known as *Peer-to-Peer* (*P2P*), is emerging. Although originally designed exclusively for pragmatic (and legally controversial) file swapping applications, Peer-to-Peer mechanisms can be used to access any kind of distributed resources and may offer new possibilities for Internet-based applications.

According to several Internet service providers, more than 50% of Internet traffic is due to Peer-to-Peer applications, sometimes even more than 75% (see also Chapter 22). The continuous growth of the Internet in terms of users and bandwidth is accompanied by increasing requirements of a diversified wealth of applications. Today, the traditional client-server approaches require a tremendous amount of effort and resources to meet these challenges. Thus, three main requirements of future Internet-based applications can be identified:

- *Scalability* is a fundamental prerequisite necessary to satisfy the vast demand for resources such as bandwidth, storage capacity, or processing power of certain applications caused by large numbers of users. Therefore, bottlenecks must be identified and avoided at an early stage of system design so the system can be scaled by several orders of magnitude without loss of efficiency.
- *Security* and *reliability* form core criteria for the availability of strategically important and security-sensitive services in the face of distributed denial-of-service attacks on centralized systems. Furthermore, anonymity and resistance to censorship are of growing importance in today's world.
- *Flexibility* and *Quality of Service* for quickly and easily integrating new services are crucial to the success of emerging Internet technologies. For example, a lack of such features prevents the wide spread deployment of highly desirable services such as group communication and mobility.

It is becoming increasingly obvious that client-server-based applications, which have become popular since the early 1980s, can no longer fully meet the evolving requirements of the Internet. In particular, their centralized nature is prone to resource bottlenecks. Consequently, they can be easily attacked and are difficult and expensive to modify due to their strategic placement within the network infrastructure. The concepts of Peer-to-Peer networking

R. Steinmetz and K. Wehrle (Eds.): P2P Systems and Applications, LNCS 3485, pp. 9-16, 2005.

and Peer-to-Peer computing[1] promise to provide simpler solutions to the problems mentioned above through a fundamental shift of paradigms.

2.1 Definitions

Oram et al. [462] gives a basic definition of the term “Peer-to-Peer” which is further refined in [573]:

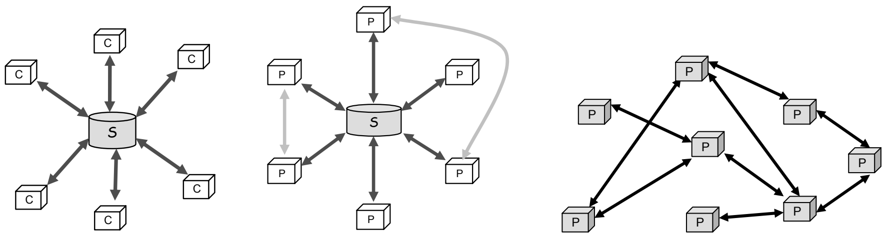
[a Peer-to-Peer system is] *a self-organizing system of equal, autonomous entities (peers)* [which] *aims for the shared usage of distributed resources in a networked environment avoiding central services.*

In short, it is *a system with completely decentralized self-organization and resource usage.* Apart from these basic principles, Peer-to-Peer systems can be characterized as follows (though a single system rarely exhibits all of these properties):

Decentralized Resource Usage:

1. Resources of interest (bandwidth, storage, processing power) are used in a manner as equally distributed as possible and are located at the *edges* of the network, close to the peers. Thus with regard to network topology, Peer-to-Peer systems follow the end-to-end argument [531] which is one of the main reasons for the success of the Internet.
2. Within a set of peers, each utilizes the resources provided by other peers. The most prominent examples for such resources are storage (e.g. for audio and video data or applications) and processing capacity. Other possible resources are connectivity, human presence, or geographic proximity (with instant messaging and group communication as application examples).
3. Peers are interconnected through a network and in most cases distributed globally.
4. A peer’s Internet address typically changes so the peer is not constantly reachable at the same address (transient connectivity). Typically, peers are dynamically assigned new Internet addresses every time they connect to the network. Often, they may be disconnected or shut down over longer periods of time. Among other reasons, this encourages Peer-to-Peer systems to introduce new address and name spaces above of the traditional Internet address level. Hence, content is usually addressed through unstructured identifiers derived from the content with a hash

[1] We do not distinguish between Peer-to-Peer computing and Peer-to-Peer networking but focus on *Peer-to-Peer (P2P)* as a property, characteristic, method, or mechanism.



(a) Client/Server (b) Hybrid (c) Peer-to-Peer

Fig. 2.1: Classification of Peer-to-Peer Systems

function. Consequently, data is no longer addressed by location (the address of the server) but by the data itself. With multiple copies of a data item, queries may locate any one of those copies. Thus, Peer-to-Peer systems locate data based on content in contrast to location-based routing in the Internet.

Decentralized Self-Organization:

5. In order to utilize shared resources, peers interact directly with each other. In general, this interaction is achieved without any central control or coordination. This represents one of the main properties of Peer-to-Peer systems which is markedly different from client-server systems: while the the latter rely on centralized coordination through a server as a structural paradigm, Peer-to-Peer systems establish a cooperation between equal partners. This departure from a centralized infrastructure most importantly avoids bottlenecks but is concomitant with the reduced availability of end-systems compared to client-server solutions.
6. Peers directly access and exchange the shared resources they utilize without a centralized service. Thus, Peer-to-Peer systems represent a fundamental decentralization of control mechanisms. However, performance considerations may lead to centralized elements being part of a complete Peer-to-Peer system, e.g. for efficiently locating resources. Such systems are commonly called *hybrid* Peer-to-Peer systems (cf. Fig. 2.1b).
7. In a Peer-to-Peer system, peers can act both as clients and servers (cf. Fig. 2.1c). This is radically different from traditional systems with asymmetric functionality (cf. Fig. 2.1a). It leads to additional flexibility with regard to available functionality and to new requirements for the design of Peer-to-Peer systems.
8. Peers are equal partners with symmetric functionality. Each peer is fully autonomous regarding its respective resources.

9. Ideally, resources can be located without any central entity or service (in Figures 2.1a and 2.1b, centralized services are necessary in contrast to Figure 2.1c). Similarly, the system is controlled in a self-organizing or ad hoc manner. As mentioned above, this guide line may be violated for reasons of performance. However, the decentralized nature should not be violated. The result of such a mix is a Peer-to-Peer system with a hybrid structure (cf. Fig. 2.1b).

2.1.1 Shift of Paradigm in Internet Communication

The Peer-to-Peer approach is by no means just a technology for file sharing. Rather, it forms a fundamental design principle for distributed systems. It clearly reflects the paradigm shift from *coordination to cooperation*, from *centralization to decentralization*, and from *control to incentives*. Incentive-based systems raise a large number of important research issues. Finding a fair balance between give and take among peers may be crucial to the success of this technology.

2.2 Research Challenges in Peer-to-Peer Systems and Applications

One important research aspect is the detailed analysis of the suitability of the Peer-to-Peer paradigm to various types of applications, in particular those beyond the domain of file sharing. During the Workshop *„Quality in Peer-to-Peer Networks"*[2] a number of types of Peer-to-Peer applications were identified and put on a time scale as illustrated by Figure 2.2. Figure 2.3 lists several possible obstacle to research in the Peer-to-Peer area and its day-to-day use. Figures 2.5 and 2.4 show certain future research challenges as found during the GI-Meetings. The main question was: When do these topics become important?

One of the main challenges of Peer-to-Peer systems lies in the decentralized self-organization of a distributed system and in achieving a high level of quality of service without the need for centralized services. Central to a solution to this problem is to efficiently look up and locate data items and to manage them accordingly. Many aspects of Peer-to-Peer systems base on this functionality. In contrast to centralized server applications, for which the location of data items is inherently known, decentralized systems store content in multiple, possibly distant, locations within the system. There are two main approaches which have been developed to solve this problem: *unstructured* and *structured Peer-to-Peer systems*.

[2] TU Darmstadt, Sept. 2003, http://www.kom.tu-darmstadt.de/ws-p2p/

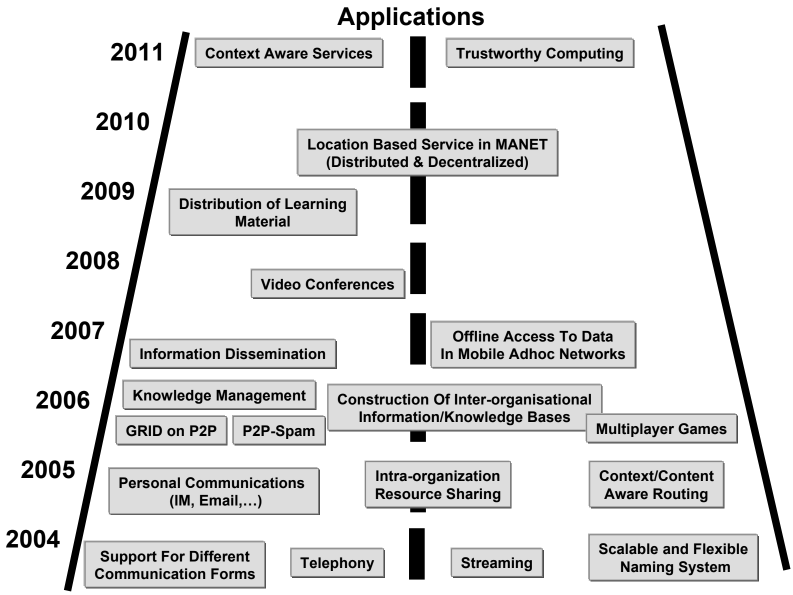


Fig. 2.2: Applications beyond File-Sharing

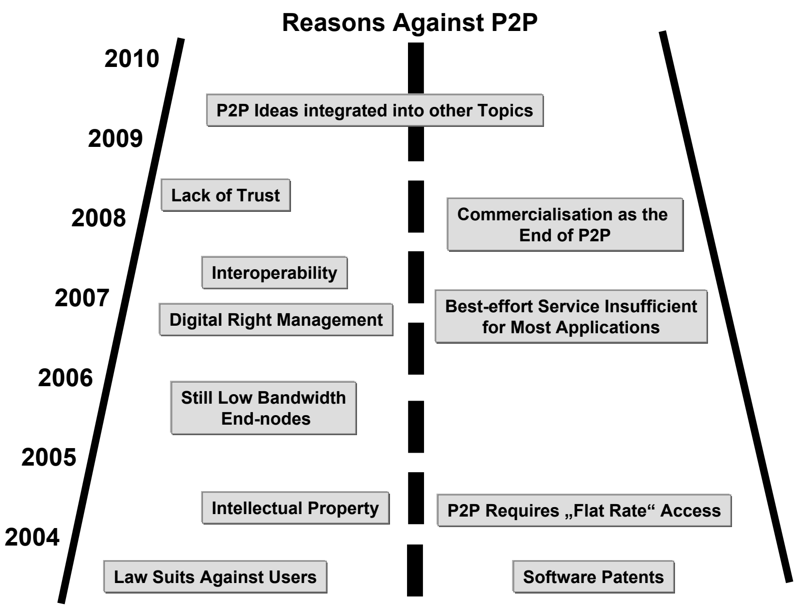


Fig. 2.3: Developments hindering the Development and Dissemination of Peer-to-Peer-Technologie.

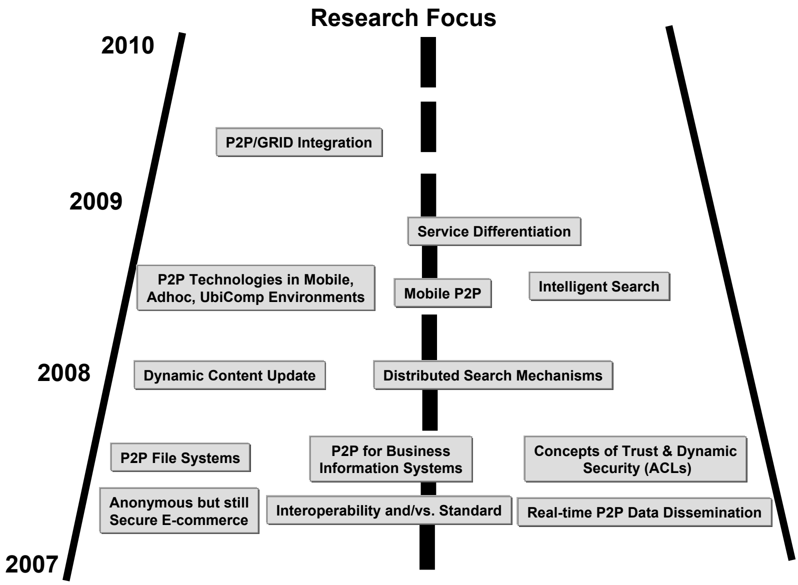


Fig. 2.4: Challenges in Peer-to-Peer Research II (2007 - 2010)

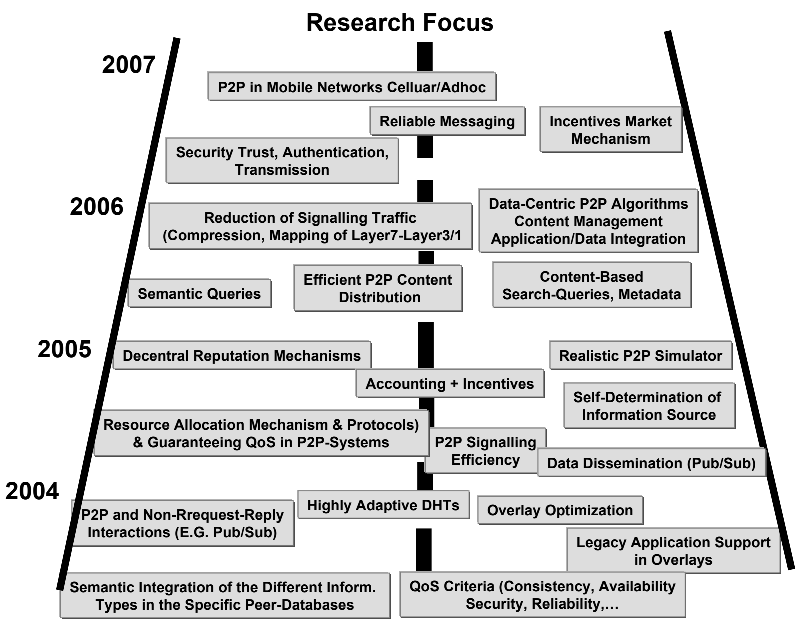


Fig. 2.5: Challenges in Peer-to-Peer Research I (2003 - 2007)

2.2.1 Unstructured Peer-to-Peer Systems

First-generation Peer-to-Peer-based file sharing applications employed so-called unstructured approaches. For example, these systems relied on lookups via a central server which stored the locations of all data items. Only after looking up the location of a data item via the server was the data transferred directly between peers (hybrid approach, cf. Fig. 2.1b). Other approaches, e.g. Gnutella, use a flooding technique, i.e. look-up queries are sent to all peers participating in the system until the corresponding data item or peer is found (cf. Chapter 5).

It is apparent that neither approach scales well. The server-based system suffers from exhibiting a single point of attack as well as being a bottleneck with regard to resources such as memory, processing power, and bandwidth while the flooding-based approaches show tremendous bandwidth consumption on the network. Generally, these unstructured systems were developed in response to user demands (mainly file sharing and instant messaging) and consequently suffer from ad hoc designs and implementations.

Part II of this book discusses unstructured Peer-to-Peer systems in more detail.

2.2.2 Structured Peer-to-Peer Systems

The challenge to develop scalable unstructured Peer-to-Peer applications has attracted the research community. Inspired by the significant advantages and possibilities of decentralized self-organizing systems, researchers focused on approaches for distributed, content-addressable data storage (distributed indexing structures). These so called Distributed Hash Tables (DHTs) were developed to provide such distributed indexing as well as scalability, reliability, and fault tolerance. DHTs outperform unstructured approaches in the above listed properties and efficiency. Commonly, a data item can be retrieved from the network with a complexity of $O(\log N)$ - equal to the complexity of well-known non-distributed search and indexing techniques. The underlying network and the number of peers in structured approaches can grow arbitrarily without impacting the efficiency of the distributed application; this is in marked contrast to the previously described unstructured Peer-to-Peer applications which usually exhibit linear search complexity at best. Necessary management operations, like adding new content or peers and handling failures, commonly have a complexity of $O(\log N)$ and $O(\log^2 N)$, respectively.

Typically, DHTs are based on similar designs, while their search and management strategies differ. Ring-based approaches such as Pastry, Tapestry, and Chord all use similar search algorithms such as binary ordered B*-trees. Geometric designs, such as *Content Addressable Networks* (CAN) or Viceroy also exist. Each peer is assigned a section of the search space $[0, 2^n - 1]$. To provide redundancy, replicas may be stored on neighboring peers. Using a routing tree, e.g. Pastry, or finger tables, e.g. Chord, a request is routed towards the desired data item. For such requests a logarithmic complexity is guaranteed. Often, the amount of routing information is in the order of $O(\log N)$ entries per peer (see also Chapter 8).

Next to the already discussed similarities to known database indexing techniques, DHTs employ additional techniques to manage data structures, to add redundancy, and to locate the nearest instances of a requested data item.

Part III in this book deals with all details on structured Peer-to-Peer systems with a special focus on Distributed Hash Tables.

2.3 Conclusion

To guarantee a wide deployment in the Internet, future distributed systems and applications must cope with several challenges. Apart from scalability, efficiency, and high flexibility, reliability and protection against attacks will form key features of future systems. Their development and successful deployment will have a strong impact on the future of Peer-to-Peer-based applications and systems.