THE APPLICATION OF THE PEER-TO-PEER NETWORK TECHNOLOGIES

Poryev G.
National Technical University of Ukraine "KPI", Kiev
e-mail: support@barvinok.net

Abtract

Poryev G. The application of the peer-to-peer network technologies. *The key differences between classic networks and peer-to-peer networks are reviewed, and their advantages and disadvantages in the applications within typical solutions are reviewed.*

Introduction

In third millennium, Internet plays very important role as the medium for data exchange and the environment for data storage. With more than two decades of development, the technologies of Internet are constantly changing, improving, becoming obsolete and replaced with new ones. The changes of basic architectural concepts for the technologies of Internet are happening much rarely. One of the most obvious and recent example of such a change is the concept of peer-to-peer networking.

The Overhead of Networks

The classical scheme of interaction and the information exchange in Internet is the well-known "client-server" concept. The server is an entity, which performs the data storage, providing the resources, such as memory, computing power, network links etc. The server does not interact directly with user or other entity requiring results. The client is an entity, which performs the requests to the server for data, processing or resources. The client also gets the response from the server and optionally displays some results to the user (see Fig.1).

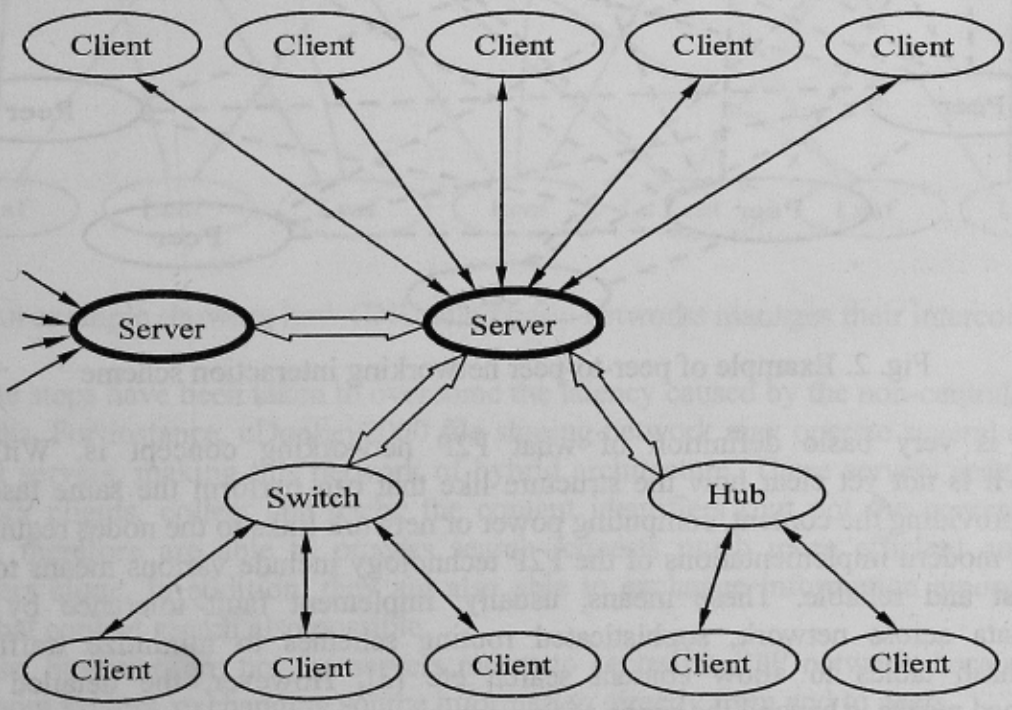

Fig. 1. Example of client-server interaction scheme

This scheme may work very well within one laboratory, one enterprise, one city and so on — until the requirements for hardware are acceptable for given quality of service. But eventually if enterprise, resource or portal have grown up to gain worldwide recognition, or to provide highly

demanded content or otherwise generate huge inflow of customers (and thus requests to the servers) the hardware becomes more and more demanding until it may eventually cross the threshold of profitability.

There are methods to relieve such an effect, like service clustering, regional mirroring, but they all are just a crouches to the originally not too flexible concept.

Peer-to-peer networking emerged to the public view early in 21st century and is questioning the very idea of separating network nodes into servers and clients.

Analysis of the researches on the P2P networks architecture

Indeed, why concentrate the content in the single point of failure, even if protected by various fault-tolerance means? Why does server need to send the very same data repeatedly to different client nodes?

The peer-to-peer (P2P) networking concept defines the participating nodes as peers, meaning equals in their relation and importance. Every node acts as a server to any other node and can act as a client requesting services from any other node. In addition, usually there is no centralized routing between any given pair of nodes — they are connecting directly "ad hoc" (see Fig.2) [1,2].

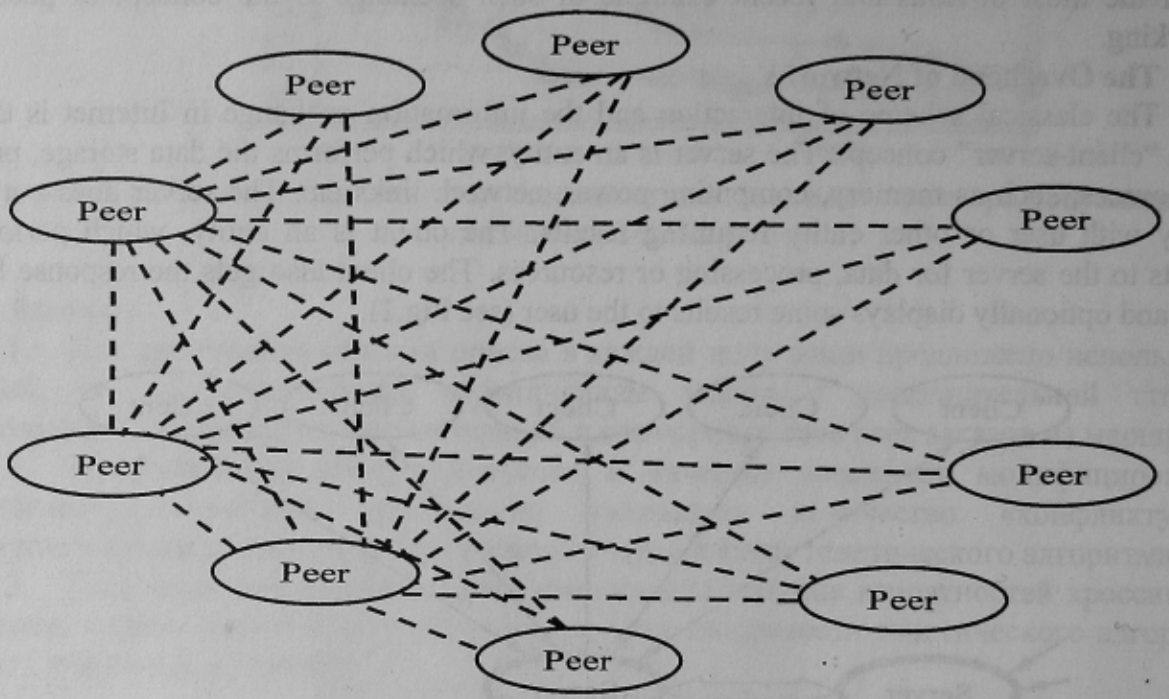

Fig. 2. Example of peer-to-peer networking interaction scheme

This is very basic definition of what P2P networking concept is. Without further explanation, it is not yet clear how the structure like that can perform the same task as "client-server", i.e. providing the content, computing power or network links to the nodes requiring it.

Most modern implementations of the P2P technology include various means to make them efficient, fast and reliable. These means, usually, implement fault tolerance by distributing redundant data across network; sophisticated routing schemes to minimize traffic overhead; distributed hash tables to allow content search etc [3]. However, the detailed analysis of aforementioned means is beyond the scope of this paper.

A Ways to Resolve

As in any evolutional process, there are always intermediate stages, which combine classic and modern approach and combine their advantages and disadvantages as well.

One of the most notable differences in using general-purpose networks based on P2P technology is their speed. For example, the file sharing networks (see below), because of their high

attractiveness, share one common feature — the demand for content is usually much higher than content's resources available at given moment. This has lead P2P network software to implement the means such as queuing of clients, using upload credit and rewarding algorithms, smart chunk distribution and so on. The speed of content search is also relatively slow in contrary to the traditional server-based resource storage, because the search request and responses must come through the several consecutive neighboring peers, multiplying at each step, while the bandwidth of these peers may have depleted at this moment.

A good example of the aforementioned feature is the GNUtella networks. They are designed to be completely decentralized, although some nodes may voluntarily choose to be a hub, depending on the bandwidth available. Unlike regular so-called "leafs", hubs are maintaining more than 2-3 connections with neighboring nodes, but this does not give them any advantage over the peers working in leaf mode.

The Research on P2P network architecture

The idea is, should one of the hubs fails, every leaf that was connected to it, also had the connection to one or two another hubs, so the overall integrity of the network is preserved. It is worth mentioning that hubs are also maintaining interconnection of some degree with more than one other hub at any time (see Fig.3).

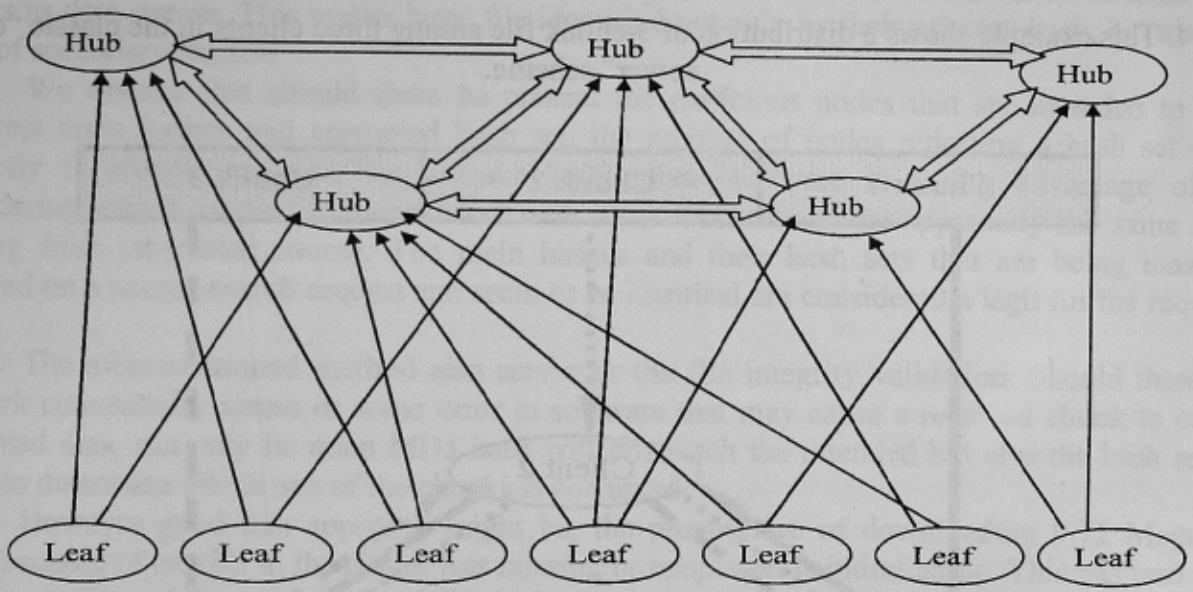

Fig.3. An example showing how GNUtella family networks manages their interconnections.

Some steps have been taken to overcome the latency caused by the non-centralized nature of P2P networks. For instance, eDonkey2000 file sharing network may operate several dedicated and independent servers, making this network of hybrid architecture. These servers accept connection from network clients, collect and cache the content identifiers (but not the content itself) from clients, and therefore are able to process search requests much more efficient and faster than network peers alone. In addition, they are also able to exchange information among themselves, making global content search also possible.

These, however, are not the servers meant to centralize P2P network, because it can still operate without servers, exchanging source information directly from and to peers.

File sharing

Today, one of the most controversial (and most heard of) applications of P2P is file-sharing networks. The idea is simple — every peer node in the network has some set of files it could share with any other peer node. A user controlling peer node can choose what files to share and what files to seek and retrieve from another peer nodes.

In the most popular file-sharing networks (eDonkey2000 and GNUtellas), the file is sliced into so-called "chunks" with the latter being distributed separately, if more than one peer nodes is requesting such a file. This slicing allows peer node containing original file to send it to network only once (in ideal case), making other nodes to share missing chunks among themselves (see Fig 4 and 5).

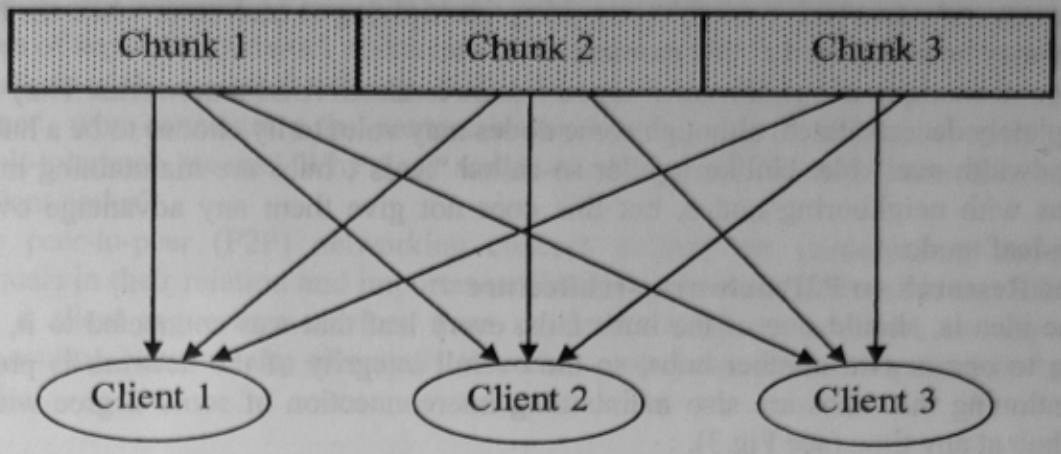

Fig. 4. This example shows a distribution of 3-chunk file among three clients in the classic "client-server" scheme.

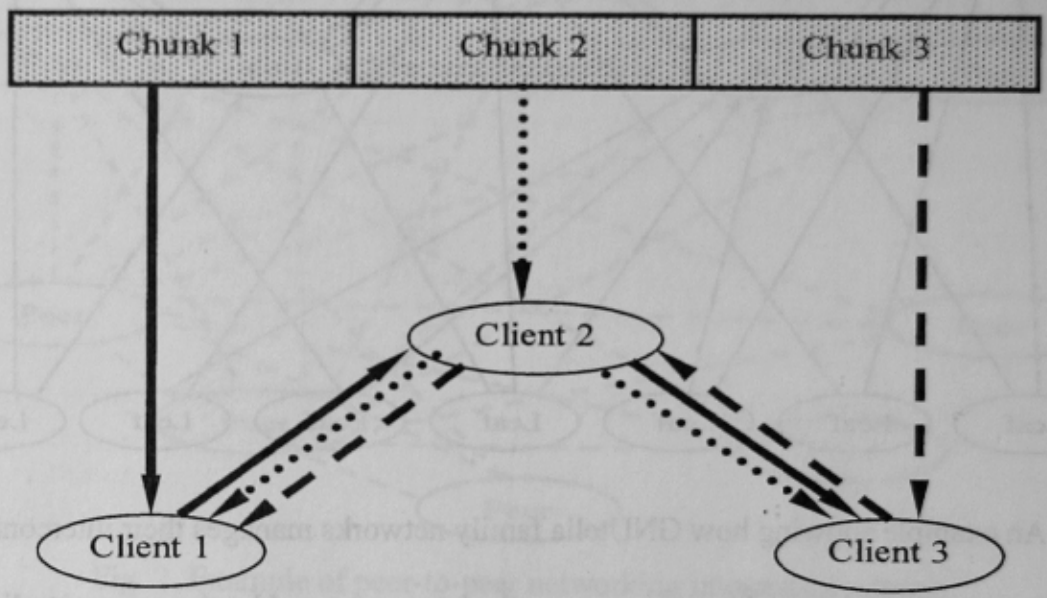

Fig. 5. This example shows a distribution of 3-chunk file among three clients in the peer-to-peer network. Note the initial distribution "server" sends each chunk only once.

While this approach could potentially save time and bandwidth to distribute some shared resource into the P2P network in ideal case, the reality is that there are usually much more requesters than chunks in a single file, and most P2P client software implements limiting the queue length for requesting peer nodes. This makes downloading some file from P2P networks rather time-consuming, despite the introduction of crediting system, where requesting peer node can advance its position in the queue by supplying chunks missing in the source node [4].

Due to its decentralized and distributed nature, it is hard (and seems to be unwanted by the majority of the uses) to implement any kind of rights management functionality — something that very early P2P concepts were vulnerable against, such as famous Napster network. This makes the modern P2P networks ideal for distributing any type of content freely, including movies, music, and software, even those protected by copyright laws. For which they are regularly become target for judicial persecution, with the most notable example being attack by Record Industry Association of

America (RIAA) via the Supreme Court of United States of America against the MetaMachine, Inc to cease and desist their operation as the founders, developers and supporters of one of the most successful P2P networks eDonkey2000.

However, this had little effect on the network as a whole, mostly because eDonkey2000 original software has about few percents of network nodes, with the majority of servers and clients are developing independently as free software.

Content validity in file sharing

Unlike the classical client-server scheme where the file is obtained from one source and therefore considered a genuine copy of the file on a server, the P2P network client receives file chunks from the multitude of network peers. Even the contents of one single chunk can be obtained from different clients. This implies the question about whether to trust the peer you are receiving parts from and how to verify if this part of file is really part you are requesting.

To address these issues, some networks, eDonkey2000 for example, has implemented certain methods.

In eDonkey2000 network, the MD5 hash identifies every file. If a file is smaller than one chunk (approximately 9.72 Megabytes), MD5 hash is taken from it alone. If a file consists of many chunks, then a separate MD5 hashes is taken from every chunk and main hash is taken from chunk hashes as data stream. This makes large file identified not only by their primary hash, but also with a set of secondary hashes.

We assume that should there be present the malicious nodes that are intended to report incorrect main hashes and corrupted hash set, the number of nodes reporting a hash set of file correctly is always greater. The eDonkey2000 client software is taking advantage of this, considering a hash set that is inconsistent with other hash sets of the apparently the same file as coming from un-trusted source. The main hashes and their hash sets that are being massively reported on a source search request and seem to be identical are considered a legit for the requested file.

The aforementioned method also serves as the file integrity validation. Should there be a network connectivity issues or some error in software that may cause a received chunk to contain corrupted data, not only its main MD5 hash will mismatch the intended but also the hash set will allow to determine which one of the chunks is corrupted.

However good this approach might be, the prospective of downloading 9.72 Megabytes again because of one bit in the chunk was missing or misplaced is intimidating. This has lead to the implementation of the so-called AICH (Advanced Intelligent Corruption Handler). AICH also works by logically splitting the file content into chunks, only the size of them in this case is just 180 kilobytes. To prevent overwhelming number of hash in hash sets derived from such a small chink, AICH employs multi-level hashing, where the next level of data consists of the hashes of previous level (see Fig.6) [5].

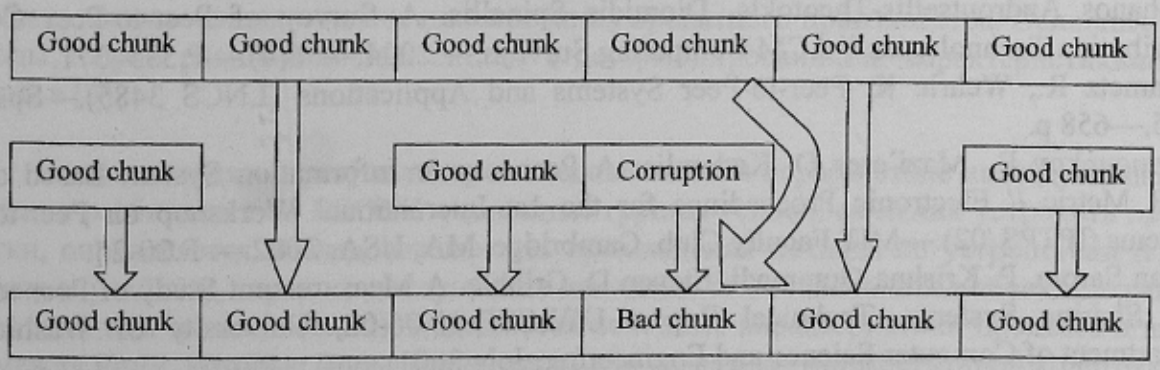

Fig 6. Corruption handling in file sharing networks

Distributed backup

Backup systems, especially those specializing on long-term archived data storage, also suffer about the same problems as classical "client-server" networking scheme. One can either spend huge investments into sophisticated hardware and software solutions, which would provide fully automated and reliable backup for the enterprise, or think about SAN or NAS-based solutions to reduce the cost by building slightly decentralized backup environment, which is still relatively expensive.

However, with P2P technology, it is possible to reduce the backup system buildout and maintenance cost to virtually the expenses for connecting dedicated backup node to the common network medium.

This idea utilizes the fact that every single node in internal network of enterprise (so called intranet) being either data processing server or workstation, never uses it's disk drive resources fully. Moreover, modern operating systems often encourage users to perform regular cleanup, in case the free space is critically low. The idea is to join all available network nodes into P2P network layer while combining some of their available disk space into the redundant storage pool, which will then logically split and used to store and retrieve backup data.

P2P network build on this principle can work if the overall size of such a pool, with the required level of redundancy taken into account, is greater than the projected amount of data it need to contain at any given time.

The security of the preserved data is achieved by implementing the encryption layer so that only originating node, which have deposited some specific data initially, can decode it to the original state.

The redundancy within this backup structure is required because not all participating nodes are available at any given moment. Some nodes on workstations can be rebooted, shut down for a while or even permanently, but P2P nature of this solution ensures that every single chunk of data is being stored at several different places. From this point of view, the inside work of distributed backup network is similar to the RAID systems.

Summary

The peer-to-peer network technologies emerged recently tend to gradually replace classical network interaction schemes, while modifying the very basic concepts of network interaction as such.

It is shown, that the application of P2P network technologies can provide highly efficient solutions at relatively low deployment and maintenance costs, especially what requires shared access to the large arrays of data.

References

1. G. Poryev. Data integrity control in the distributed networks // Eastern European Magazine on Advanced Technologies.—2006.—№ 4/2 (22).—p.32-35
2. Stephanos Androutsellis-Theotokis, Diomidis Spinellis. A Survey of Peer-to-Peer Content Distribution Technologies // ACM Computing Surveys,— 2004.—36(4).—P.335–371.
3. Steinmetz R., Wehrle K. Peer-to-Peer Systems and Applications (LNCS 3485).—Springer, 2005.—658 p.
4. Maymounkov P., Mazi`eres D. Kademlia: A Peer-to-peer Information System Based on the XOR Metric // Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02).—MIT Faculty Club, Cambridge, MA, USA, 2002.— P.20-25.
5. Stefan Saroiu, P. Krishna Gummadi, Steven D. Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. Technical Report UW-CSE-01-06-02, University of Washington, Department of Computer Science and Engineering, July 2001..