LECTURE SERIES

Peer-to-Peer Networks

1

Lin Yu[1], *Cheng Shiduan*[1], *Li Qi*[2]
(1. Beijing University of Posts and Telecommunications, Beijing 100876, China;
2. Peking University, Beijing 100088, China)

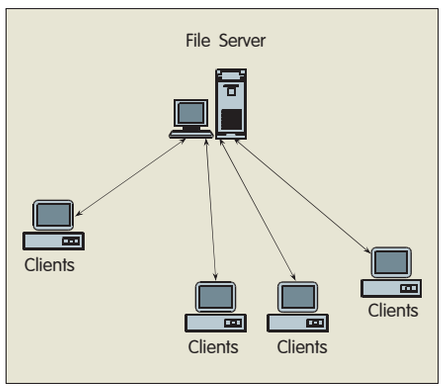
The development of network resources changes network computing models. P2P networks, a new type of network adopting peer-to-peer strategy for computing, have attracted world-wide attention. P2P architecture is a type of distributed network in which all participants share their hardware resources and the shared resources can be directly accessed by peer nodes without the necessity of going through any dedicated servers. The participants in a P2P network are both resource providers and resource consumers. This article on P2P networks will be divided into two issues. In this issue, P2P architecture, network models and core search algorithms are introduced. And the second part in the next issue will analyze the current P2P research and application situations, as well as the impact of P2P on telecom operators and equipment vendors.

Peer-to-Peer (P2P) networks adopt a peer-to-peer strategy for computing. However, the conventional Internet computing models were dominated by client/server architecture. Several years ago, the network bandwidth was relatively narrow and computing resources were poor on the client end. Therefore, client/server architecture could centralize processing activities at servers, decreasing the requirements on terminal capabilities. In recent years, however, different resources present different development speeds: network traffic doubles every 6 months; network bandwidth (transmission rate in core fiber network) doubles every 7 months; the development of computing resources basically follows Moore's Law (i.e., doubles per 18 months); and the storage capability is only raised by 7% per year. Accordingly, computing and storage resources would become the bottlenecks of network development, and the central server of network architecture would become the bottleneck of network performance. The whole service system would collapse once the central server breaks down. Under such situations, the P2P computing model is introduced.

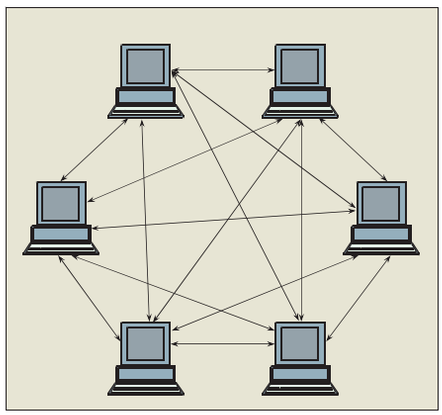
With the development of terminal and network access technologies, terminals have more powerful capabilities. P2P architecture uses the collaboration of terminals at the edge of the network to avoid possible bottlenecks in network performance caused by centralized architecture. P2P architecture breaks the conventional client/server model. In a P2P network, each node has equivalent capabilities and responsibilities. That is, each node not only works as a server to provide other nodes with service, but also enjoys service offered by other nodes. client/server and P2P architecture is shown in Figure 1 and Figure 2 respectively.

P2P architecture has different definitions in different industries. A typical definition is that P2P architecture is a type of distributed network in which all participants share their hardware resources (such as processing capabilities, storage capabilities, network access capabilities and printers) and the shared resources can be directly accessed by peer nodes with no necessity of going through any dedicated servers. The participants in a P2P network are both resources (service and contents) providers and consumers.

In this article, a chess-playing application system is used as a simple example to help readers understand some basic principles of P2P architecture. In conventional client/server architecture, the chess-playing system consists of two types of nodes: the chess-playing server and chess players. The working process of the system is as follows: Player A and Player B log in to the chess-playing server, and the server then matches them to play on a chessboard. Every step A plays is implemented by the message procedure of "A-the chess-playing server-B".



▲ Figure 1. Client/Server architecture.



▲ Figure 2. P2P Architecture.

Both control management flow (users' log-in and player matching) and service flow (chess playing process) for each chess program need the participation of the server. Therefore, a chess-playing system with millions of simultaneous users requires a server group with a large number of servers and powerful capabilities.

P2P technology has the following characteristics:

(1) Decentralization

The resources and services in a P2P network are distributed to all the nodes. Transmission of data and implementation of services are conducted directly between nodes with no intermediate links and servers, which avoids possible bottlenecks. Still using the chess-playing system as an example: the service flow of chess playing goes directly between two nodes of players, and the central server is unnecessary (except the services that need centralized management such as billing and scoring). Decentralization is a basic feature of the P2P network, which brings the network with scalable and robust capabilities.

(2) Scalability

When more users log in to a P2P network, the whole resources and service capabilities of the system are improved (because new chess users themselves also offer services and resources) even though the demands on services are increased. Therefore, the demands of users may be well met. The whole system is distributed without any obvious bottlenecks. Take the chess-playing system as an example. The service capabilities (including chessboard creation and management of playing rules) are mainly provided by the player nodes, and accordingly the chess-playing server is less burdened.

(3) Robustness

P2P architecture has the advantages of attack and error tolerance. A P2P network is generally built up by self-organization, and allows any nodes to join and leave it freely. Different P2P networks adopt different topologies. They may keep adjusting their topologies according to the changes of network bandwidth, the number of nodes and the load. Since services are implemented between nodes, the breakdown of some nodes or a part of the P2P network will have little influence on other nodes or other parts of the network (that is, the breakdown of a network for two P2P chess players won't influence others' chess playing). Even if some nodes fail to work, the P2P network may keep the inter-connection of other nodes in the network by a topology self-adjustment.

(4) High Performance/Cost Ratio

P2P architecture can make effective use of numerous ordinary nodes distributed throughout the Internet. It distributes computing tasks and data for storage to all the nodes, and makes full use of idle computing capabilities and storage spaces to reach the goal of high performance computing and massive storage. A P2P chess-playing system won't need as many servers, because a large part of service is shared by the user nodes.

(5) Privacy Protection

In P2P networks, the transmission of data is conducted among all the nodes with no need to go through any servers for centralized management. This greatly lowers the possibility of wiretapping or leakage of users' privacy. Currently, the main solution to privacy protection on the Internet is to use the relay transfer to hide the communication participants in numerous network entities. In a conventional anonymous communication system, privacy protection relies on certain relay servers (for instance, billing and scoring in a conventional chess-playing system are implemented through the central server). On the other hand, all the participants on the P2P network may support relay transfer, which greatly improves the flexibility and reliability of anonymous communications, and accordingly protect users' privacy better. However, this advantage of the P2P network is also its weakness, for it is usually used by illegal organizations to transport private messages (for example, it is easier for chess players to cheat on a P2P chess-playing system, because there is no central server to conduct supervision).

1 P2P Topology Structures

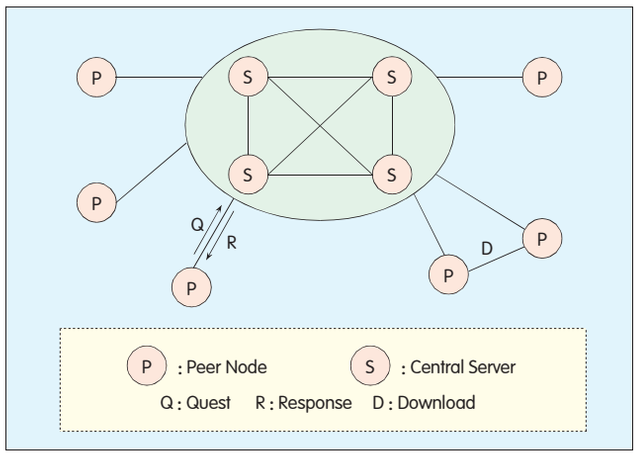
Topology refers to the physical or logic interconnection between computing units in a distributed system. The topology of nodes is always an important basis on which the type of a system is defined. The popular topologies on the Internet include centralized, and layered topologies. The centralized topology is facing certain problems such as excessive storage and Denial of Service (DoS) attacks. P2P architecture has decentralized topologies that can be of the centralized, decentralized unstructured, decentralized structured or partially decentralized type. The main challenges of P2P topologies include naming and organizing numerous nodes in the system, defining the join/leave modes of nodes, implementing error recovery, and more.

1.1 Centralized Directory Structure

The centralized directory P2P structure (the centralized topology) is the earliest P2P application model. It is also called non-pure P2P topology because of its centralization feature. Its typical application is Napster, well-known MP3 sharing software.

Napster uses a central server to store the index of uploaded music files and data about the storage location. When a user wants a music file, he first accesses and searches the Napster server, and the sever will send him back the information about the user who has that file. Then the user with the quest will directly access the owner of the file to download it. In the Napster model, a group of high-performance central servers stores the directory information about the shared resources of all P2P computers. When there is a file lookup quest, a peer computer will send the quest to a central server. The central server searches and lists addresses of qualified peer computers for it. After receiving the response of the central computer, the computer that initiated the quest will, according to network traffic, time delay, and other conditions, choose a suitable peer computer to build up a connection and download the file. Figure 3 shows the topology and working principles of Napster.

Take the chess-playing application as an example. The central server of Napster's architecture fulfils management services such as the log-in and match creation of players functions. However, once two players begin to play



◀Figure 3.
Model adopted by MP3 sharing software Napster.

chess, the central server won't interfere with the game play process that is individually completed by the two players. Therefore, the essence of Napster is the separation of file searches (management service) and file transmission (specific services). This effectively saves the resource consumption of the central server. This type of topology has the following strengths:

(1) Maintenance is simple.

(2) Resource finding is efficient.

(3) Finding algorithms are flexible and efficient, and complex searches can be implemented since the resource lookup relies on a centralized directory system.

However, this model still has many problems.

(1) The most dangerous risk is the central server. Since Napster still adopts a centralized structure for file searches, the whole system will break down once the central server fails to work (just like chess playing cannot keep going once the chess–playing server breaks down). When the number of users is raised to 105 or more, the performance of the Napster system will degrade severely. The breakdown of the central server easily causes the collapse of the whole system, and therefore its reliability and security are poor.

(2) With the expansion of the network, the cost for maintaining and updating central search servers will increase greatly.

(3) The existence of the central server may cause troubles in copyrights of shared resources.

Therefore, with strengths in management and control, the centralized directory model is applicable to small networks, rather than large–scale networks.

1.2 Pure P2P Network Model

Decentralized unstructured topology uses a random graph in overlay networks. Its typical application is the Gnutella system. Gnutella is a P2P file sharing system. Compared with the Napster system, it is a pure P2P system without any search servers. In Gnutella, each node randomly maintains its local topology. As shown in Figure 4, Gnutella adopts completely–random–graph–based flood finding and random walker. When a node searches for some information, Gnutella sends a broadcasting message to surrounding nodes to query them about searched information. If one of the surrounding nodes has information, it directly sends it to the searching node. Gnutella uses decrement of the Time To Live (TTL) to control the transmission scope of information searching.

Take the chess–playing application as an example. There is no centralized chess–playing servers in a decentralized unstructured system. If a player wants to play chess, he directly queries surrounding nodes about who will play chess with him (topology knowledge of the surrounding nodes is obtained randomly, such as the person who played chess with him, or who queried him). If there is a volunteer, the service matching is fulfilled. If no one wants to play with him, the surrounding nodes will continue to query the nodes around themselves until a volunteer is found or the matching fails.

With Gnutella, every online computer is equal in functionality. They are both clients and servers, and accordingly called Servents. The number of nodes on the Internet follows Power–law, that is to say, a few nodes are involved in most node connection cases. This causes the phenomenon of the little world. (Two strangers may communicate with each other through six middlemen at most. As for the chess–playing application, the volunteer can be found after a repeated query in most cases.) Therefore, Gnutella can quickly find destination nodes, and has a high tolerance to face the dynamic changes of the network.

However, under a situation in which nodes connected to the Gnutella network keep increasing and the network keeps expanding, the flood finding will cause a sharp increase of network traffic (for example, when many chess players

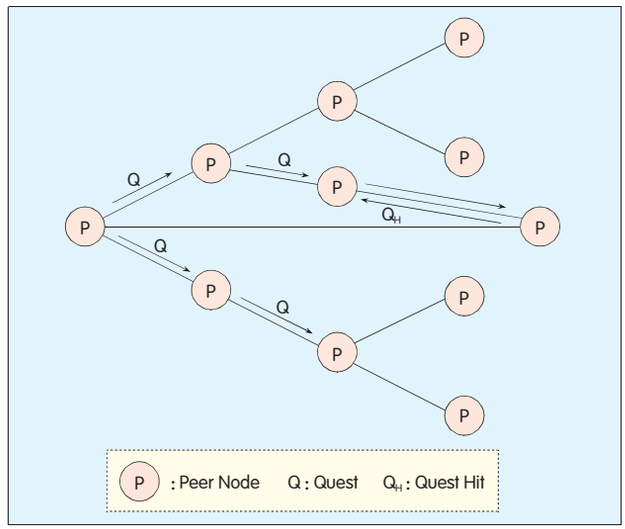


Figure 4.▶
The flooding search algorithm adopted by Gnutella.

query the players surrounding them at the same time, it is easy to sound an avalanche–like loud noise). Accordingly, some nodes with narrow bandwidth in the network will break down due to overload. Therefore, severe segmentation and link breaking existed in the primary Gnutella network (that is, the user groups of Gnutella could not implement entire interconnection).

The exactness of resource finding and scalability are two important challenges to unstructured networks. Similar to Gnutella, the FreeNet system adopts a decentralized model, but it has some improvements. Although Gnutella and FreeNet support a decentralized search strategy, both of them adopt a flooding mechanism that is similar to the Open Shortest Path First (OSPF) routing protocol. (In fact, an OSPF system itself is also a P2P network.) A flooding mechanism may not only cause a heavy burden of network communication, but also has bad scalability. With similar situations, the OSPF protocol is accordingly only used for Autonomous Systems (AS) of the Internet.

Generally, unstructured networks cannot offer performance guarantees, and their search results may be incomplete. The system adopting a broadcasting search consumes a massive amount of network bandwidth, which causes bad scalability and other problems. In order to solve these problems, much research is focused on how to build a highly structured system. The network model based on decentralized structured topology discussed in the next section is this type of system.

1.3 Structured Network Model

The basic difference between so–called structured and unstructured models is whether the neighbors a node maintains are organized by some special rules that are applicable to the entire network, or organized randomly. The former organization mode fulfills the rapid search between nodes.

Structured P2P is a model for location service, which adopts a purely distributed message transfer mechanism and searches according to key words. The Distributed Hash Table (DHT) is a leading technology for this model.

DHT is actually a huge hash table commonly maintained by a large number of nodes in a wide area. The hash table is segmented into discontinuous sections, and each node is assigned a hash section. The node is the manager of its hash section. With DHT technology, each node is given a unique node ID in some way. A resource object creates a unique resource ID by hashing. (In the chess–playing application, each player has a unique ID through which the surrounding nodes of a player can be communicated by certain a algorithm. In this way, all players are organized as a loop). When the resource object is searched, the node having it may be found through hashing.

Four classical DHT application cases are introduced here. They are Chord, Content Addressable Networks (CAN), Pastry and Tapestry.

The most important contribution the Chord algorithm makes is to give a distributed lookup protocol. This protocol maps designated key words to corresponding nodes. When the Chord algorithm is used for a network composed of N nodes, it is unnecessary for each node on the network to know data about all the other nodes, and what it needs to do is to maintain data about other $O(\log N)$ nodes. Therefore, each lookup only needs $O(\log N)$ messages. When there is a node joining or leaving the network, the algorithm needs to update routing data. Each log–in or log–out needs to send $O(\log_2 N)$ messages.

The CAN algorithm uses multidimensional identifier space to implement distributed hash. CAN maps all the nodes to an n–dimensional Descartes space, and distributes each node segment as evenly as possible. The routing algorithm CAN uses is direct and simple: when the coordinates of the object node are known, the quest is sent from the current node to one of its surrounding nodes that has coordinates nearest to the object node.

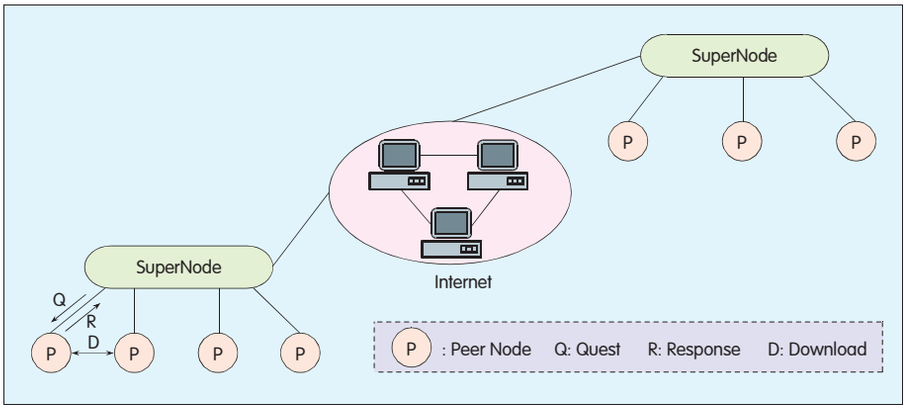
Pastry is a scalable distributed objective location and routing algorithm proposed by Microsoft Research. It can be used for large–scale P2P systems. Pastry distributes to each node a 128–bit node ID. All the node IDs form a ring, ranging from 0 to $2^{128}-1$. When a new node joins in the system, its IP address is randomly assigned in the 128–bit space by hashing.

The Tapestry algorithm comes from the Plaxton algorithm. In Plaxton, a node uses the neighboring node table it knows to transfer messages step by step according to the destination ID. Based on Plaxton, Tapestry adds an error tolerance mechanism, and accordingly adapts to dynamic changes of the P2P system.

With structured topology, DHT architecture is adaptive to the dynamic log–in/log–out of nodes. Besides, it has good scalability, robustness, even distribution of node IDs, and a self–organization capability. DHT helps offer accurate resource finding. It can always find a destination node as long as this node is on the network. In general, DHT is applicable to large–scale peer–to–peer network applications. At present, this technology is mainly applied to data and files sharing systems.

The biggest problem of DHT architecture is its complex maintenance mechanism, while network churning caused by the frequent log–in and log–out of nodes will greatly raise the cost of DHT maintenance (just like in the chess–playing application, each log–in and log–out of players brings an adjustment of P2P topology). Moreover, based on accurate hash, DHT only supports an exact key words matching lookup. It cannot support a complex lookup such as content or semantic lookup. This is another problem. Take the chess–playing application as an example: Player lookup fulfills only through IDs, but some fuzzy information such as the found player's rating and performance of network connected to him cannot be used for lookup.

DHT–based routing mechanism also has inextricable problems. For example, after hashing, the location information of nodes is destroyed. Therefore, IDs of the nodes from the same sub–network are possibly far away from each other. This goes against the optimization of lookup. (DHT–based topology may not match with actual network topology. For example, it is possible that two chess players from Beijing and Shanghai respectively are assigned adjacent locations, but they actually have a great time delay when communicating on



▲Figure 5. Partially decentralized topology (with SuperNode).

▼Table 1. Comparison of 4 P2P structures

Compared Items \ Topology	Centralized Topology	Decentralized Unstructured Topology	Decentralized Structured Topology	Partially Decentralized Topology
Scalability	Bad	Bad	Good	Middle
Reliability	Bad	Good	Good	Middle
Maintainability	Best	Best	Good	Middle
Algorithm Finding Efficiency	Highest	Middle	High	Middle
Complex Search	Supporting	Supporting	Not Supporting	Supporting

the network.)

1.4 Hybrid Network Model

As shown in Figure 5, Kazaa is a typical P2P hybrid model (the partially decentralized structure). It introduces the concept of SuperNode into the pure P2P distributed structure, integrating fast lookup of centralized P2P and decentralization of pure P2P.

Kazaa divides nodes into ordinary nodes and search nodes (in some cases, nodes are divided into three categories) according to node capabilities (such as computing capability, memory, bandwidth and staying time). A search node and some ordinary nodes around it compose an autonomic cluster. Each cluster has a centralized P2P structure, and different clusters are connected by a pure P2P structure. Further, the node with best performance out of the search nodes, or a new introduced node with best performance may even serve as an index node to store data about available search nodes on the entire network and to maintain the whole network structure.

For an ordinary node, it first searches files in the cluster to which it belongs. Only when the search results are not substantial, is limited flooding made between search nodes. In this way, network congestion, slow lookup and other disadvantages brought by the flooding algorithm in the pure P2P structure are effectively avoided. Moreover, the search node in each cluster monitors all ordinary nodes in the same cluster, which ensures that some malicious attacks are controlled on local networks. In addition, SuperNodes, to some extent, improve the load balance of the entire network.

Generally speaking, SuperNode based hybrid P2P architecture has many improvements. However, frangibility of SuperNodes may cause isolation of ordinary nodes in the same cluster. Therefore, this partial search has its limitation, which leads to the emergence of a structured P2P network model.

The partially decentralized structure has strengths of good performance, good scalability and easy management. However, relying on SuperNodes, it is prone to be attacked, and its error tolerance is also influenced.

1.5 Comparison of 4 P2P Structures

Table 1 compares the comprehensive performance of the 4 P2P structures discussed above. They have different balances of system complexity, scalability and functionality.

(to be continued)

Manuscript received: 2005-11-22

Biographies

Lin Yu, PhD, is now an associate professor at the National Lab of Switching and Networking of Beijing University of Posts and Telecommunications. His research interests include control, measurement and management of QoS, P2P technologies and mobile applications. He has taken part in more than ten projects sponsored by the National "973" Program, the National Natural Science Foundation of China and the National "863" Program as a senior researcher or project leader. He has published more than 30 papers in IEEE Trans. on Neural Networks, IEEE Trans. on Wireless Communications, Journal of High-speed Networks, IEEE InfoCom, GlobeCom, ICC, WCNC and more. He also owns 18 patent applications.

Cheng Shiduan is a professor and doctoral advisor at Beijing University of Posts and Telecommunications. She is also the vice director of the academic committee of the University. Her current research is in ISDN, ATM, TCP/IP, voice communication technologies for ATM and IP networks, protocol engineering, traffic engineering, broadband network performance and QoS, etc..

Li Qi is a professor and doctoral advisor at Peking University. She is also the director of the technical committee of the China Society of Image and Graphics, a member of the National Geographic Information Standardization Technical Committee of China, a member of the National RFID Standards Working Group of China, and is a specially invited professor of the Data and Information Center of State Oceanic Administration of China. In addition, she is a consultant to the Informationization Office of the State Council of China, an expert consultant to the Beijing Municipal Government, and an expert on China Digital Earth Development Strategy Research.