Peer-to-Peer (P2P) Computing

Yi Zhang

Agenda

- History
- What is P2P
- Client/Server and P2P
- Why P2P
- Problems and possible solution
- P2P middleware services
- References

History

- Predecessors: instant messaging, whiteboarding
- Revolution of computing model for the Internet
- Napster:started in September 1999, more than 20 million users by mid-2000
- SETI@home: more than 2.6 million users, over 500,000 years of processor system time by early 2001
- October 2000, first meeting of the Peer-to-Peer Working Group, which was formed by Intel, HP, Hitachi, Fujitsu, etc.

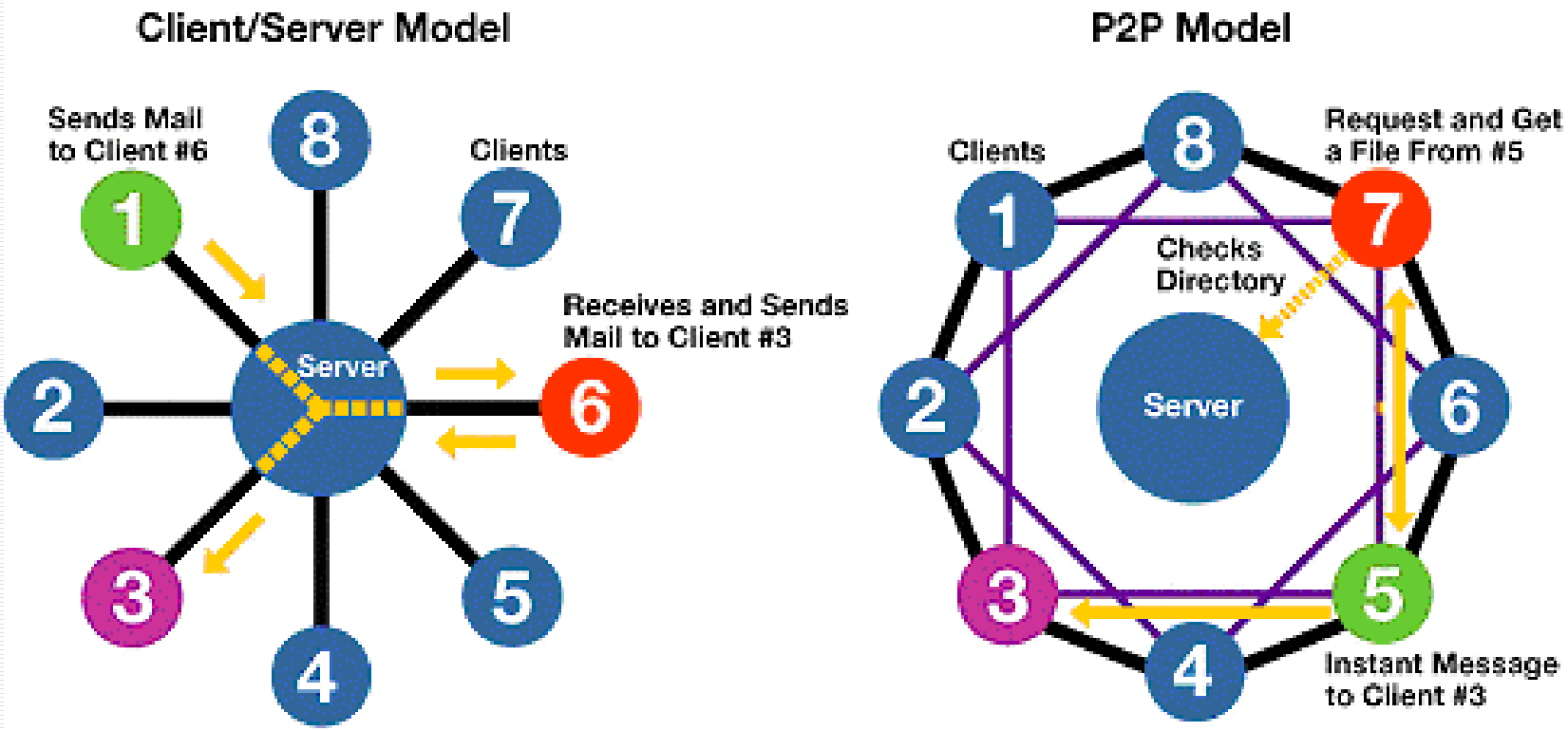
What is P2P

- Some have called peer-to-peer computing the third generation of the Internet.
- P2P neither begins nor ends with Napster.
- P2P is hardly new. Some old hands argue that it's exactly what the Internet is and always has been about. Many of the Internet's elements are peer to peer, such as file transfer and Telnet for remote logon.
- P2P is a mindset, not a particular technology or an industry. P2P provides a new way of utilizing distributed resources, which we say are found at the edge of network. Since typical networks consist of centralized servers, many people call this mindset "decentralization“.

What is P2P (cont)

Definition: P2P computing is the sharing of computer resources and services by direct exchange between systems. These resources and services include the exchange of information, processing cycles, cache storage, and disk storage for files. Peer-to-peer computing takes advantage of existing desktop computing power and networking connectivity, allowing economical clients to leverage their collective power to benefit the entire enterprise.

Client/Server and P2P



Client/Server and P2P (cont)

- P2P is an alternative to the traditional client/server architecture.
- While employing the existing network, servers, and client infrastructure, P2P offers a computing model that is orthogonal to the client/server model.
- The two models coexist, intersect, and complement each other.

Client/Server and P2P (cont)

- Peer functions as a client with a layer of server functionality, it may act both as a client and as a server within the context of a given application.
- P2P users are liberated from the traditional dependence on central servers and they have a higher degree of autonomy and control over the services they utilize.
- One of the greatest benefits of P2P computing is community. P2P makes it possible for users to organize themselves into ad hoc groups that can efficiently and securely fulfill requests, share resources, collaborate, and communicate.

Why P2P

Technical advantages

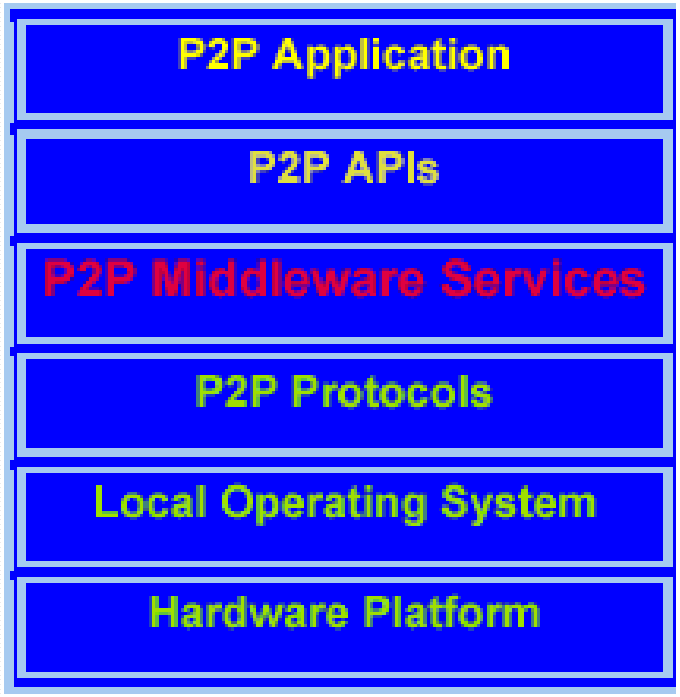
- Make use of vast untapped resources that go unused without it.
- Elimination of the single-source bottleneck
  - distribute data and control and load-balance requests across the net
  - eliminate the risk of a single point of failure
  - in enterprise, replace some costly data center functions with distributed services between clients
  - direct access and shared space => remote maintenance capability

- Social and psychological factors
  - Autonomous online communities
  - Ability to bypass centralized control

Problems and possible solution

- Problems
  - No templates and wizards are available to build applications
  - Lack of reusable components
  - Core protocols, libraries and APIs to enable platform independence and P2P application interoperability are still missing
- P2P middleware services
  - Place of P2P middleware services in the system
  - Services provided

P2P middleware services



The Place of P2P Middleware Services

P2P Middleware Services

P2P middleware services: Overview

- Standards
  - In most of these seven areas there are a variety of standards and common industry specifications.
  - An important point is to use widely adopted standards wherever possible.
  - Promote ease of development
  - Cooper with other types of networking applications
- Policies
  - Most of the services in the seven layers are mechanisms (for communication, sharing, discovery, etc.)
  - Different applications using the same mechanism may not be able to work together if they are using different policies.
  - P2P middleware services must include specification of policies, method for discovering policies or methods for negotiating common policies.

P2P middleware services: Communications

- No permanent network address => name space and directory capability
- Support communications in presence of network discontinuities such as firewalls and NAT (Network Address Translators), should be transparent to the application developers
- Communicate when peers are not online simultaneously => store-and-forward mechanisms
- One peer causes code to run on another => the issues of authentication, verification and possible encryption of data must be addressed by a robust security architecture
- The implementation and protocols of the Communication layer should be based on common well-specified industry specifications => TCP/IP, HTTP, XML, and SOAP etc

P2P middleware services: Availability

- Assurance that alternative resources will be available, even in the event some peer in the network becomes unavailable
- Intermittent connectivity => store and forward mechanism
- Resuming sharing resources at a later time
- Determine which capabilities a cooperating application provides
- Event and Exception Management Services
  - event and exception propagated to other peers, and actions taken
  - services to discover and recover from failures that affect cooperating P2P applications
  - publish/subscribe model of interaction

P2P middleware services: Security

- Security mechanisms
  - Security must be designed into the architecture, and not implemented as an afterthought.
  - Services: Identification, authentication, confidentiality, integrity, attestation and availability
  - Example: use of standard SSLs for message transport
- Policies

P2P middleware services: Identity, Presence, Community

- Peer: person, not a computer. Ability to identify and authenticate a user.
- User and Group Management Services
  - Linked to the notification services to enable a publish/subscribe model to the group, thus enabling group notification
  - Link group identity with network services to enable multicasting of data to the group.
  - Combine “group identity” with security services to enable an “access control list”
  - Support a user to carry his identity from one system to another
  - Support continuing a user session from one device to another, like that in Microsoft’s Passport service

P2P middleware services: Administration, Monitoring

- Relatively high level services that connect to many of the basic requirement on the middleware services.
- The variety of administration and monitoring capabilities needed in various situations is quite daunting => likely the best solution is to provide a wide range of data access controls and administrative hooks.

P2P middleware services: Naming, Discovery and Directory

- Naming: To name all the objects we meet: peer computers, sharable resources, network resources, users, accounts, groups, services
- Discovery and Directory: locate certain objects or objects meeting certain requirements, and report the relevant associations between different named objects.

P2P middleware services: Sharable Resources

- Storage Services: provide a programming abstraction for storage in a peer-to-peer application, and mask the complexity of dealing with persistent storage on multiple operating systems across a network boundary.
- File Services: provide traditional file I/O semantics to the programmer. Usually an abstraction on top of storage services.
- Object Management Services: ensure the availability of objects requested by remote peers, and ensure the deactivation/destroying of an object when it is no long being used.
- Processor Services: provide the programmer with scheduling semantics that allow distribution of work across many heterogeneous computers.

References

- http://cedar.intel.com/cgi-bin/ids.dll/topic.jsp?catCode=BYM
- http://www.p2pwg.org/whatis/index.html
- http://www.oreilly.com/catalog/p2presearch/summary/
- http://cedar.intel.com/media/pdf/Vision_0_9_2.pdf
- http://felter.org/wesley/p2p/one/P2PInfrastructure.html