The distributed systems discussed in Chapter 2 allow access to resources in a tightly controlled environment.

System administrators enforce security rules and control the allocation of physical rather than

virtual resources. In all models of network-centric computing prior to utility computing, a usermaintains

direct control of the software and the data residing on remote systems.

This user-centric model, in place since the early 1960s, was challenged in the 1990s by the peer-topeer

(P2P) model. P2P systems can be regarded as one of the precursors of today’s clouds. This new

model for distributed computing promoted the idea of low-cost access to storage and central processing

unit (CPU) cycles provided by participant systems; in this case, the resources are located in different

administrative domains. Often the P2P systems are self-organizing and decentralized, whereas the

servers in a cloud are in a single administrative domain and have a central management.

P2P systems exploit the network infrastructure to provide access to distributed computing resources.

Decentralized applications developed in the 1980s, such as Simple Mail Transfer Protocol (SMTP), a

protocol for email distribution, and Network News Transfer Protocol (NNTP), an application protocol

for dissemination of news articles, are early examples of P2P systems. Systems developed in the late

1990s, such as the music-sharing system Napster, gave participants access to storage distributed over

the network, while the first volunteer-based scientific computing, SETI@home, used free cycles of

participating systems to carry out compute-intensive tasks.

The P2P model represents a significant departure from the client-server model, the cornerstone of

distributed applications for several decades. P2P systems have several desirable properties [306]:

• They require a minimally dedicated infrastructure, since resources are contributed by the participating

systems.

• They are highly decentralized.

• They are scalable; the individual nodes are not required to be aware of the global state.

• They are resilient to faults and attacks, since few of their elements are critical for the delivery of

service and the abundance of resources can support a high degree of replication.

• Individual nodes do not require excessive network bandwidth the way servers used in case of the

client-server model do.

• Last but not least, the systems are shielded from censorship due to the dynamic and often unstructured

system architecture.

The undesirable properties of peer-to-peer systems are also notable: Decentralization raises the

question of whether P2P systems can be managed effectively and provide the security required by

various applications. The fact that they are shielded from censorship makes them a fertile ground for

illegal activities, including distribution of copyrighted content.

In spite of its problems, the newparadigmwas embraced by applications other than file sharing. Since

1999 new P2P applications such as the ubiquitous Skype, a Voice-over-Internet Protocol (VoIP) telephony

service,5 data-streaming applications such as Cool Streaming [386] and BBC’s online video service, content distribution networks such as CoDeeN [368], and volunteer computing applications

based on the Berkeley Open Infrastructure for Networking Computing (BOINC) platform [21]

have proved their appeal to users. For example, Skype reported in 2008 that 276 million registered

Skype users have used more than 100 billion minutes for voice and video calls. The site

www.boinc.berkeley.edu reports that at the end of June 2012 volunteer computing involved

more than 275,000 individuals and more than 430,000 computers providing a monthly average of

almost 6.3 petaFLOPS. It is also reported that peer-to-peer traffic accounts for a very large fraction of

Internet traffic, with estimates ranging from 40% to more than 70%.

Many groups from industry and academia rushed to develop and test new ideas, taking advantage

of the fact that P2P applications do not require a dedicated infrastructure. Applications such as Chord

[334] and Credence [366] address issues critical to the effective operation of decentralized systems.

Chord is a distributed lookup protocol to identify the node where a particular data item is stored. The

routing tables are distributed and, whereas other algorithms for locating an object require the nodes to

be aware of most of the nodes of the network, Chord maps a key related to an object to a node of the

network using routing information about a few nodes only.

Credence is an object reputation and ranking scheme for large-scale P2P file-sharing systems. Reputation

is of paramount importance for systems that often include many unreliable and malicious nodes.

In the decentralized algorithm used by Credence, each client uses local information to evaluate the reputation

of other nodes and shares its own assessment with its neighbors. The credibility of a node depends

only on the votes it casts; each node computes the reputation of another node based solely on the degree

of matching with its own votes and relies on like-minded peers. Overcite [337] is a P2P application to

aggregate documents based on a three-tier design. The Web front-ends accept queries and display the

results while servers crawl through the Web to generate indexes and to perform keyword searches; the

Web back-ends store documents, meta-data, and coordination state on the participating systems.

The rapid acceptance of the new paradigm triggered the development of a new communication

protocol allowing hosts at the network periphery to cope with the limited network bandwidth available

to them. BitTorrent is a peer-to-peer file-sharing protocol that enables a node to download/upload large

files from/to several hosts simultaneously.

The P2P systems differ in their architecture. Some do not have any centralized infrastructure, whereas

others have a dedicated controller, but this controller is not involved in resource-intensive operations.

For example, Skype has a central site to maintain user accounts; users sign in and pay for specific

activities at this site. The controller for a BOINC platform maintains membership and is involved in

task distribution to participating systems. The nodes with abundant resources in systems without any

centralized infrastructure often act as supernodes and maintain information useful to increasing the

system efficiency, such as indexes of the available content.

Regardless of the architecture, P2P systems are built around an overlay network, a virtual network

superimposed over the real network. Methods to construct such an overlay, discussed in Section 7.10,

consider a graph G = (V, E), where V is the set of N vertices and E is the set of links between them.

Each node maintains a table of overlay links connecting it with other nodes of this virtual network,

each node being identified by its IP address. Two types of overlay networks, unstructured and structured,

are used by P2P systems. Random walks starting from a few bootstrap nodes are usually used by

systems desiring to join an unstructured overlay. Each node of a structured overlay has a unique key that

determines its position in the structure; the keys are selected to guarantee a uniform distribution in a very large name space. Structured overlay networks use key-based routing (KBR); given a starting node

v0 and a key k, the function KBR(v0, k) returns the path in the graph from v0 to the vertex with key k.

Epidemic algorithms discussed in Section 7.12 are often used by unstructured overlays to disseminate

network topology.