Peer-to-Peer Architecture

Daniel Briskin
Lohit Talasila
Natalie Ho

What is P2P?

Unlike server-client architecture where there is a clear distinction in the roles of the client and the server, in peer-to-peer architecture all nodes (peers) function simultaneously as both clients and servers. Peers communicate directly with each other, rather than connecting to a central authority.

Vocabulary

- **Components**
  - Peer (or peer node): A member of the network, functions as both a client and a server

- **Connectors**
  - P2P protocols: An established format of communication among nodes. Examples include BitTorrent, Gnutella, Tor, I2P, Bitcoin, Ethereum
Why is P2P architecture useful?

Due to the fact that there does not exist a central authority in the network, the solution to the single point of failure problem is inherent in the design of the architecture. As such, P2P architecture allows for the creation of a distributed networks where no one node is of critical importance to the network as whole.

Unlike client-server architecture, P2P architecture becomes more powerful as the number of users increases. In client server architecture, the increase in requests from users will lower the availability of the server and may eventually cause the server to crash. However in P2P architecture, the amount of resources available to the network increases as more peers join. This makes P2P architecture extremely scalable.

The decentralized nature of P2P architecture makes these networks very difficult to dismantle. Since there is no central server through which everyone must communicate, dismantling a P2P network would require each node to be taken down individually. To carry out this task on a healthy network would be a very costly pursuit.

Some examples of P2P architectures include BitTorrent (file sharing), Tor (anonymous internet browsing), and Bitcoin (a decentralized transaction ledger).

Topological Constraints

Peers only have access to the data that they share with each other. In order for a peer to be able to access a piece of data, there must exist another peer currently on the network possessing that piece of data. Additionally, there must exist a path of accessible nodes between the two peers so that a connection may be formed.

Given this limitation, when a peer first connects to a network, it must rely on its immediate neighbours to share information on the existence of other nodes. As such, it takes time for a peer to establish a strong connection to a network.

It is important to note that a P2P network is not necessarily a complete graph; some nodes may be connected to all (or most) nodes in the network, while others may be connected to only a small subset of peers. Some nodes may have as few as one other immediate neighbour. In the event that two nodes are not directly connected, they may still exchange information if there exists a path of nodes between them.

Applications

- Censorship resistant networks and communication
- Distributed computation platforms
- Trustless systems
Relevant Non-Functional Properties

● Supports:
  ○ Availability/Dependability/Reliability
    ■ The decentralized nature of the P2P network architecture removes the possibility of having a single point of failure. When a peer goes offline, the network as a whole is not compromised and may continue to function as normal. Healthy P2P networks are capable of sustaining 100% uptime.
  ○ Inexpensive
    ■ Peers share their resources with the network, the architecture does not require the initial invest of purchasing a central server.
  ○ Scalability
    ■ The more peers there are in a network, the healthier it is. The system improves as it grows.
  ○ Performance
    ■ A healthy P2P network with a lot of available resources can perform at speeds equivalent to or greater than an expensive central server setup.
  ○ Democratic
    ■ Since there is no central authority in the network, it is up to the peers themselves to determine what changes they wish to incorporate into the protocol and what data they are willing to provide access to. The opinion of an individual peer does not influence the network as a whole; it is only through mutual agreement among many peers that a change can be made to the protocol.

● Inhibits:
  ○ Security
    ■ P2P network architectures are largely unsecured since the typical workstation will run a standard desktop operating system that may not provide the security features to be tolerant to attacks on the P2P architecture. Also the fact that P2P applications act as servers as well as clients makes them more vulnerable to remote exploits. Security largely depends on the specific implementation of a P2P protocol; not all protocols are equal.
  ○ Reliability
    ■ Users in a P2P network are completely responsible for deciding what content is available. Some files will eventually disappear and become unavailable as more people stop sharing them.
  ○ Serviceability
    ■ Given the decentralized nature of P2P architecture, it is difficult to coordinate a protocol change across all peers. A set of peers may choose to ignore an upgrade, or may even choose to modify the protocol to serve their own purposes. This can lead to network fragmentation.
  ○ Performance
Unhealthy P2P networks can be slow and unresponsive, to the point where it is not even usable.

- Regulation
  - It is up to the peers themselves to decide how the P2P network is used. It is next to impossible to control what data is being shared across the network.

**Change Resilience**

Due to the decentralized nature of P2P networks, the absence of a single point of failure allows for the network to freely grow and shrink as peers join and leave. The network’s resources increase whenever a peer joins, thus allowing the system scale freely to accommodate the higher load. As there is no single point of failure, a P2P network may only be dismantled by shutting down each node individually. Given these characteristics, P2P is a form of architecture that is very resilient to change.

However, P2P networks are not resilient to changes in protocol. Since there is no central authority to define the protocol that must be used, if a group of peers choose to adopt a new version of the protocol, they may not be able to communicate with the rest of the network. This makes updating P2P protocols problematic; if the update is not well coordinated, network fragmentation may occur.

**Weaknesses**

- Routing attacks: The network can be disrupted by malicious peers deliberately sharing invalid data in order to confuse others.
- Lack of moderation: It is hard to control the data that is being shared, and thus it is hard to stop illegal activity.
- Unreliability when accessing unpopular data: Sharing files in a P2P network requires that at least one node in the network has the requested data, and that the provider node is able to connect to the node requesting the data.

**Demonstration**

- Sticky notes represent resources, such as data, disk storage or processing power.
- Nodes A, B and C represent peers in the P2P network.

**Scenario 1**

Node A has a blue, a red, and a yellow sticky note.
Node B has nothing.
Node C has nothing.

B: request blue!          A gives B a blue sticky note.
C: request yellow!        A gives C a yellow sticky note.
B: still need a red and yellow! A gives B a red, C gives B a yellow.
C: still need a red and blue! A gives C a red, B gives C a blue.

This demonstrates the basic features of P2P, where all nodes can communicate with each other. A node can provide and receive help from other nodes at the same time (acting like server and client). C received help from both A and B, and B received help from both A and C. Node A was not required to send all three files to both B and C, as B and C were able to share with each other.

Scenario 2

Node A has blue and red sticky notes. B and C have nothing.

B & C: request blue and red.
A gives red to B and blue to C.
Now A lost connection.

B: still looking for blue.
C: still looking for red.
C gives B blue. B gives C red.

This demonstrates that in P2P no one node is of critical importance to the network. The failure of A will not affect the connection between B and C. There is no single point of failure.

Scenario 3

A has 3 red sticky notes (the first half of a huge file).
B has 2 red sticky notes (the second half of a huge file).

C: request the entire file (5 red sticky notes). A and B send sticky notes to C simultaneously.

This demonstrates in a P2P network, huge files/work can be broken down into smaller pieces. Each node can store/process a part of it, and when needed they can reassemble the entire file.

Scenario 4

A has red sticky notes but doesn't bother going online.
B requests red sticky notes but can never get it.

This demonstrates that in P2P users are completely responsible for deciding what content is available.

Scenario 5

A is infected by a virus (green sticky notes), and will constantly wants to share green sticky notes with its neighbor.
B & C request blue sticky notes but got green sticky notes from A instead. The virus would spread very quickly because of the nature of P2P. This demonstrates the vulnerability of P2P. Once a virus starts to spread in a P2P network, it will be very difficult to take down.

Scenario 6

Node A and Node B are connected and Node B is connected to Node C. Note that Node A and Node C are not directly connected.

A has a blue sticky note and B and C have nothing
B goes online and requests a blue sticky note. A sends a blue sticky note to B
C goes online and requests a blue sticky note. B sends a blue sticky note to C

Note that in this scenario though both A and B have a blue sticky note at the time when C requests for it, but only B is able to transfer it to C as they are connected. This scenario demonstrates that the P2P network graph need not be well connected (i.e., a complete graph), in this case it's a line graph.