Are distributed peer-to-peer overlay networks worth the effort?

Jacob Thebault-Spieker

University of Minnesota, Morris

Dec. 4, 2010

Outline

1. Definitions
   - Peer-to-Peer Network
   - Overlay Network
2. Why Do We Care?
3. Types of Overlay Networks
   - Unstructured Distributed Peer-to-Peer Overlay Networks
   - Hybrid Distributed Peer-to-Peer Overlay Networks
   - Structured Distributed Peer-to-Peer Overlay Networks

Acronym

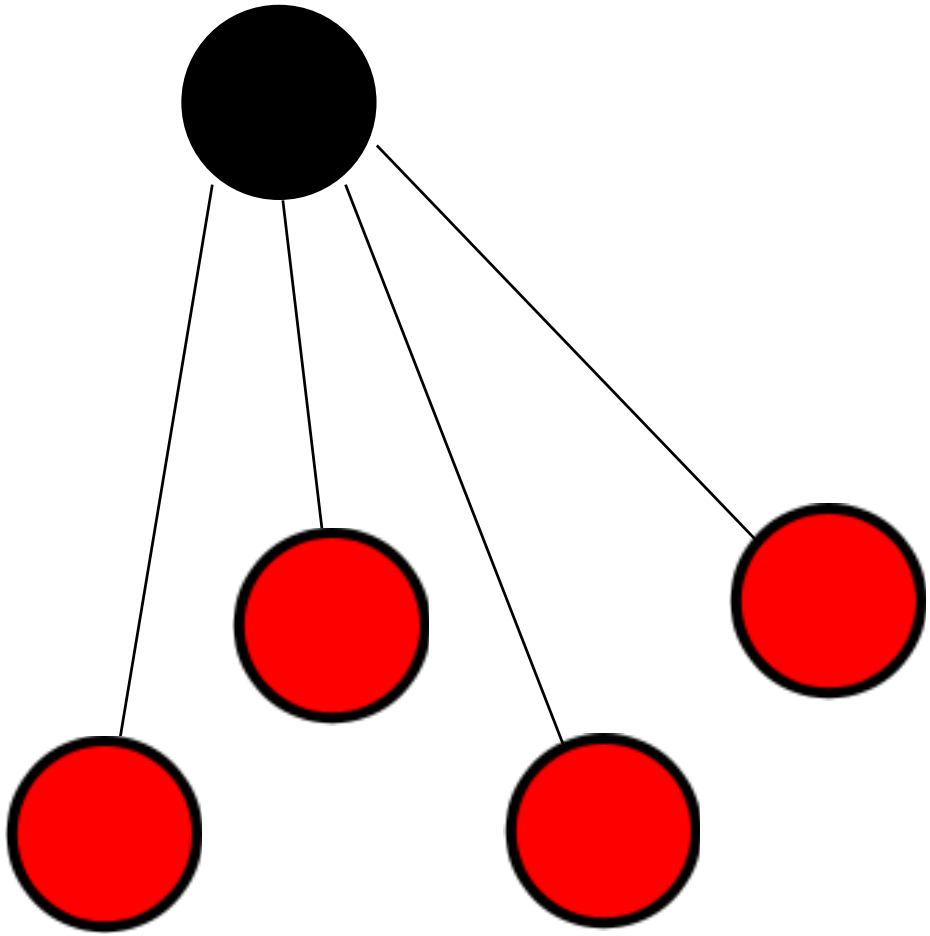
For the purposes of speed and clarity, we will refer to distributed peer-to-peer overlay networks as dP2P overlay networks throughout this presentation.

Outline

Peer-to-Peer Network

Peer-to-peer networks

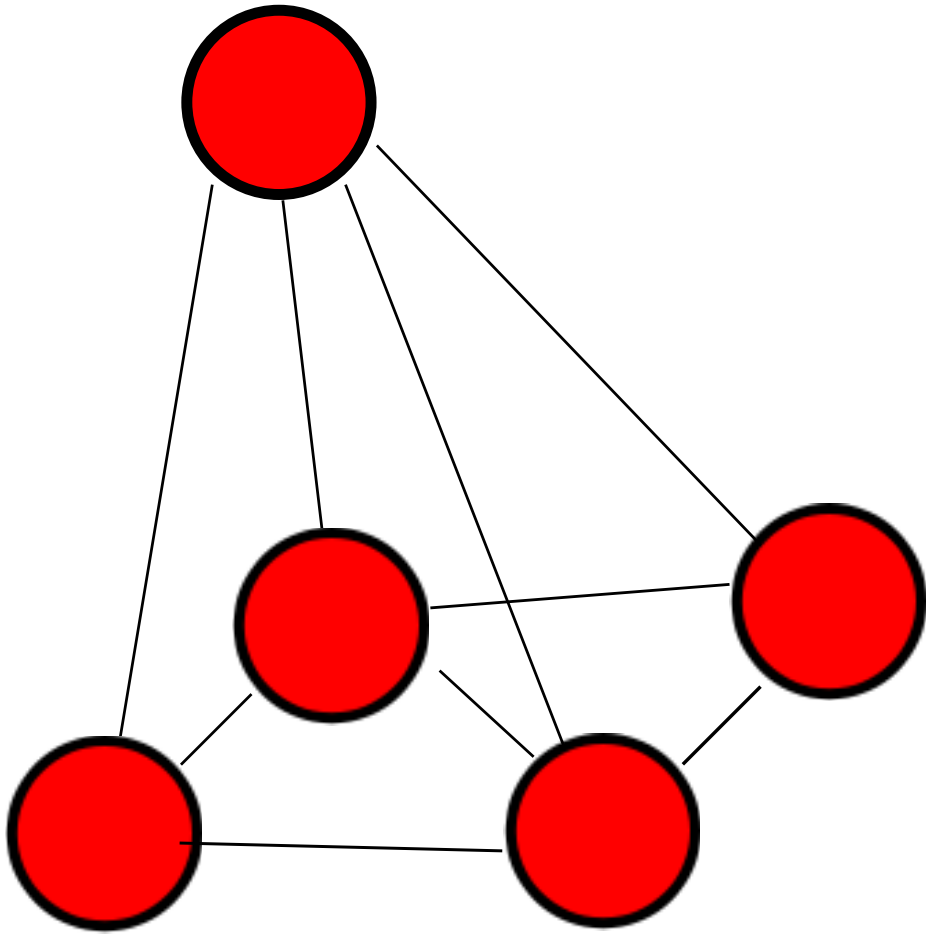
- do not have a centralized server.
- allow direct interaction between peers (or nodes).



Peer-to-Peer Network

Peer-to-peer networks

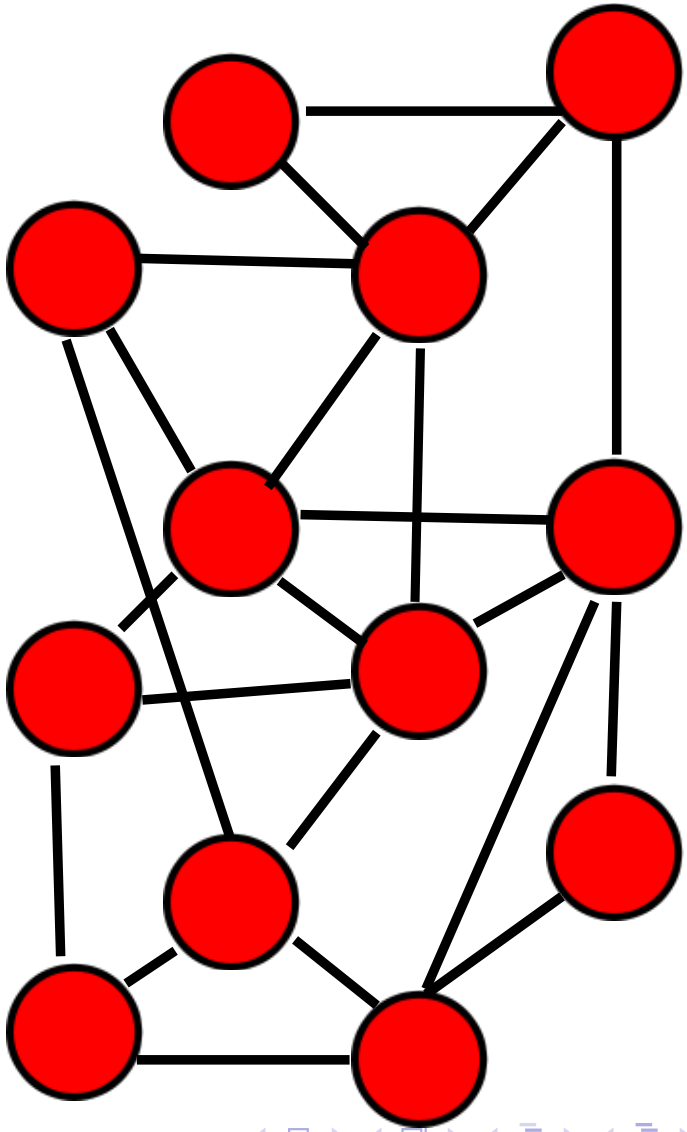
- do not have a centralized server.
- allow direct interaction between peers (or nodes).



Overlay Network

Overlay networks have three primary characteristics:

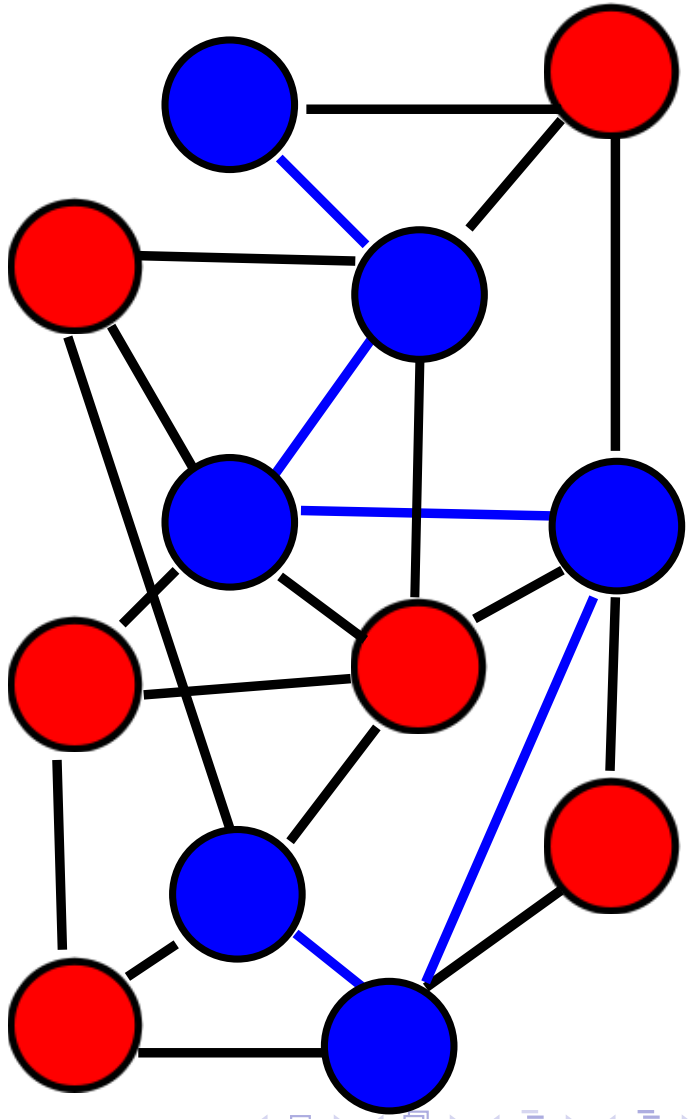
- independent of physical infrastructure
- “laid over” a traditional network
- sub-set of total machines on the network



Overlay Network

Overlay networks have three primary characteristics:

- independent of physical infrastructure
- “laid over” a traditional network
- sub-set of total machines on the network

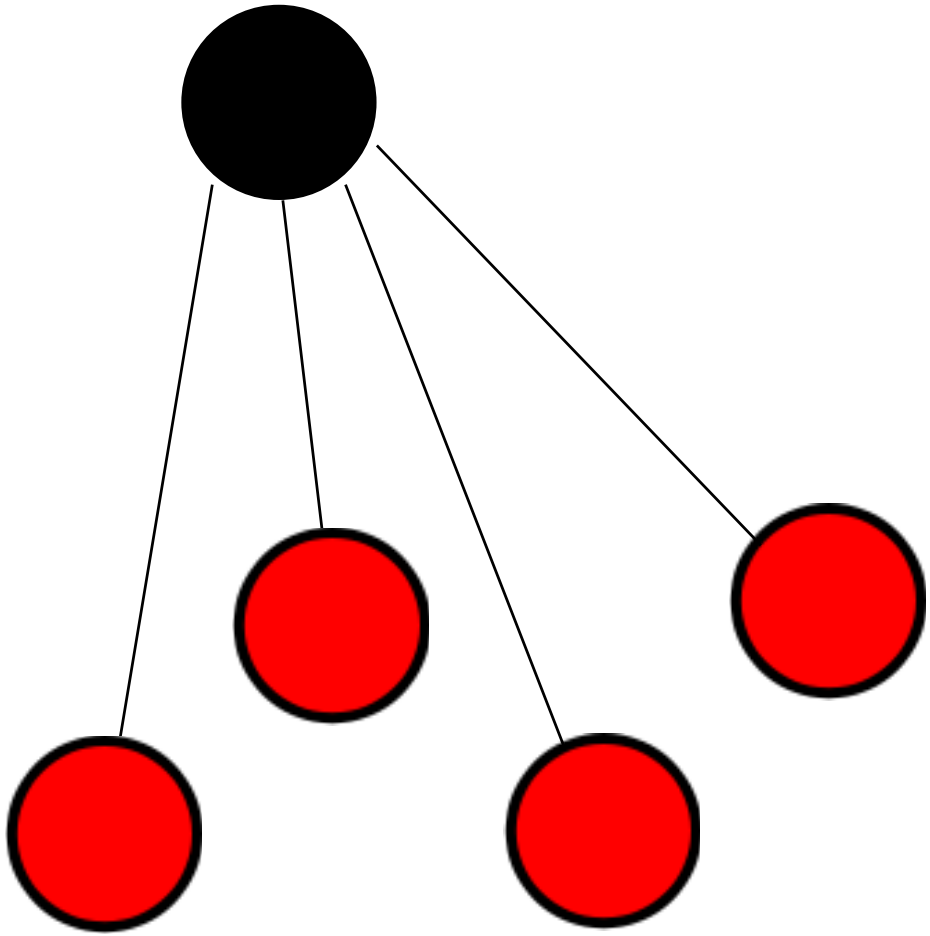


Outline

Benefits of Distributed Peer-to-Peer Overlay Networks

The primary benefits of this type of network are:

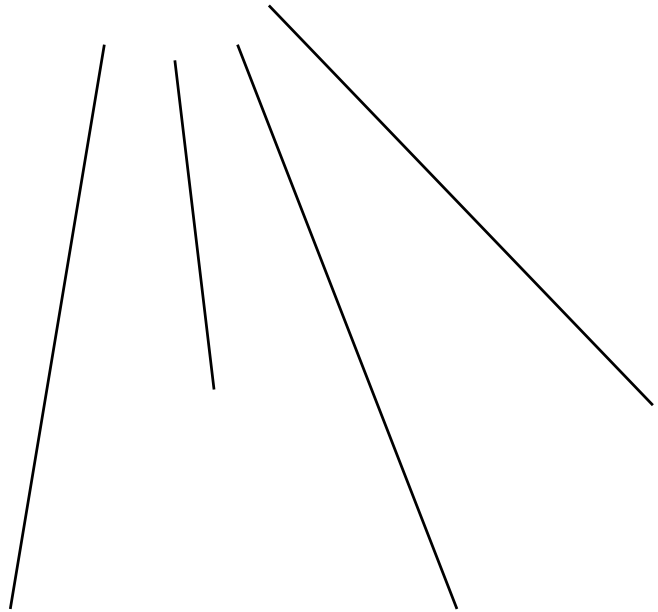
- reliability
- decentralization
- no need for additional infrastructure



Benefits of Distributed Peer-to-Peer Overlay Networks

The primary benefits of this type of network are:

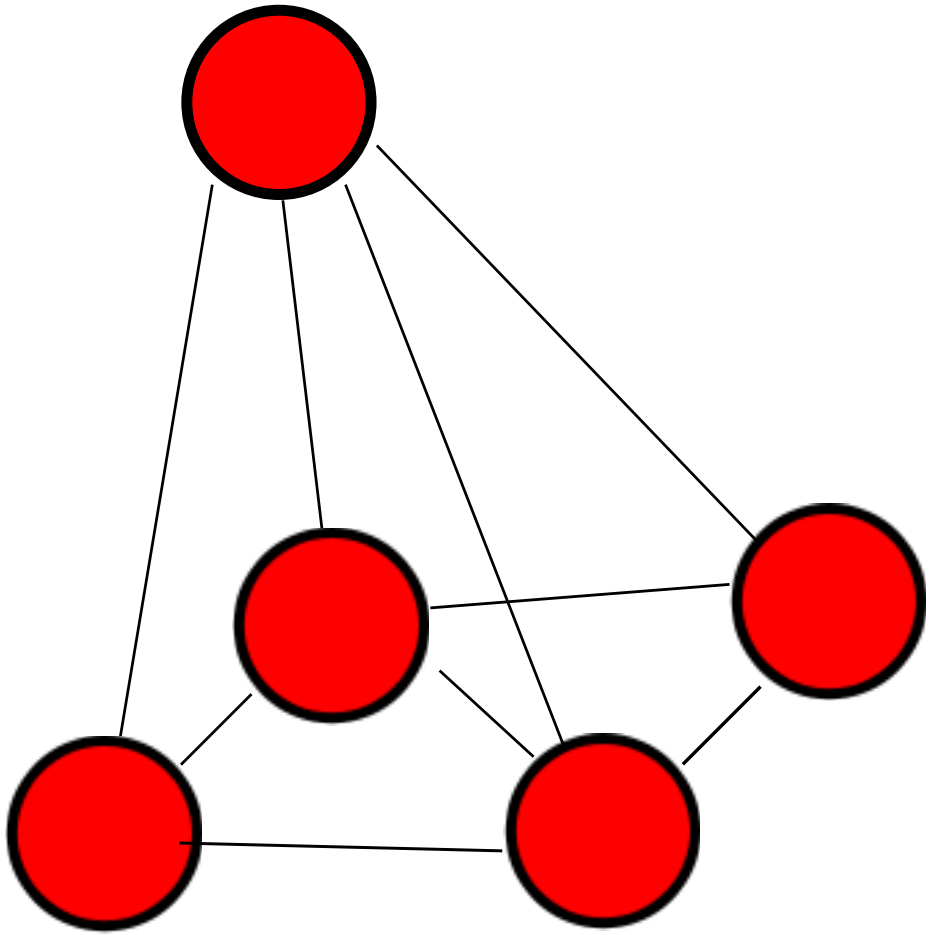
- reliability
- decentralization
- no need for additional infrastructure



Benefits of Distributed Peer-to-Peer Overlay Networks

The primary benefits of this type of network are:

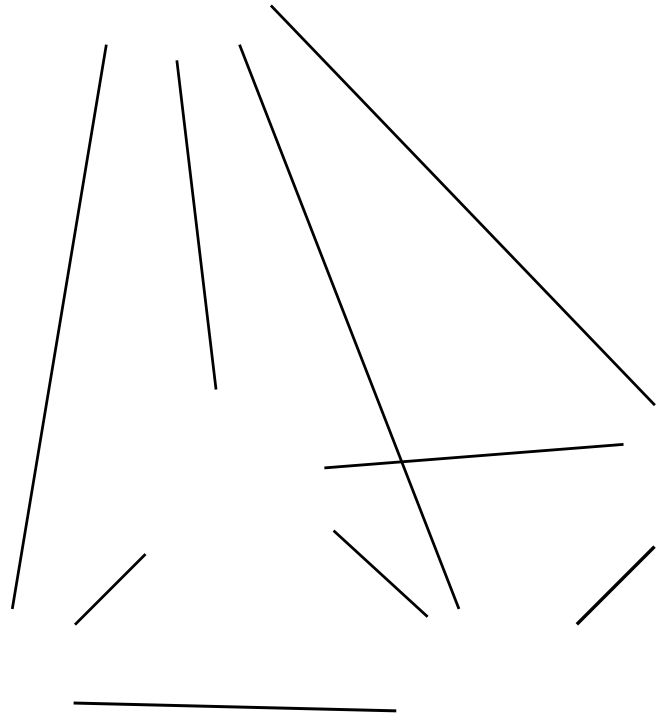
- reliability
- decentralization
- no need for additional infrastructure



Benefits of Distributed Peer-to-Peer Overlay Networks

The primary benefits of this type of network are:

- reliability
- decentralization
- no need for additional infrastructure



Outline

Unstructured dP2P overlay networks

Unstructured overlay networks allow direct communication between peers.
Benefits:

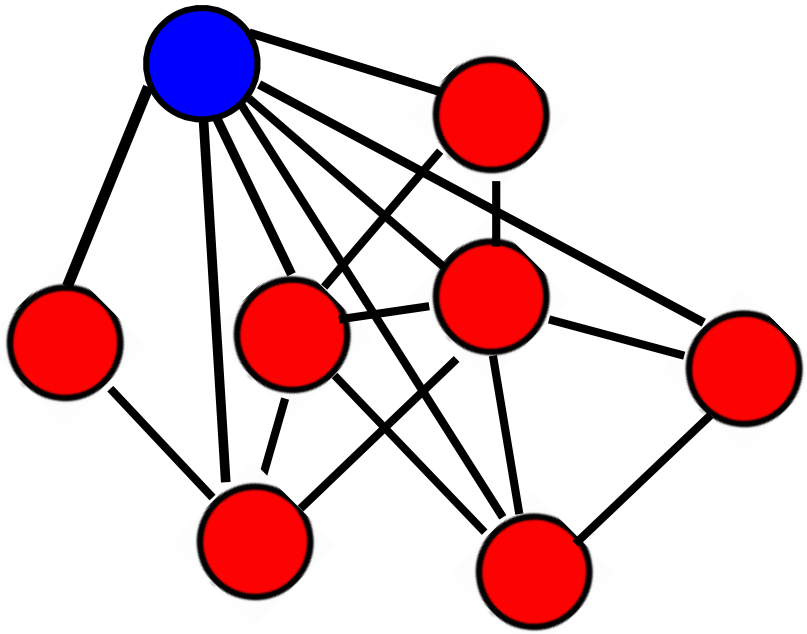
- not dependent on a single server
- unaffected when a node joins or leaves

Downsides:

- search is not guaranteed to complete
- popular items get replicated often
- unpopular items do not

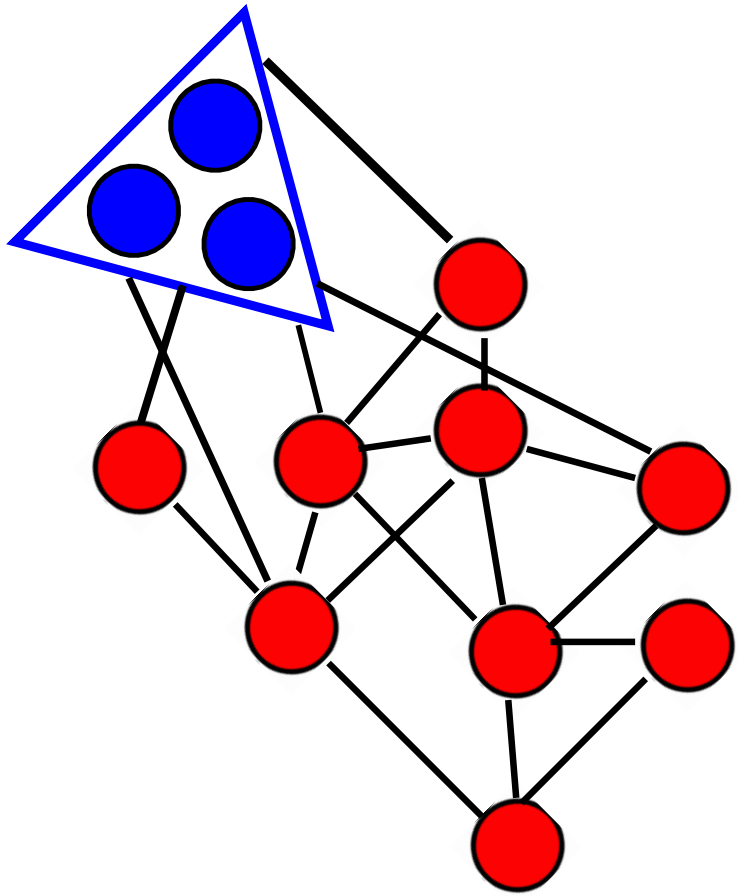
Hybrid dP2P overlay networks

Hybrid Overlay networks attempt to deal with the issues of Unstructured dP2P Overlay Networks by introducing 1+ lookup servers.



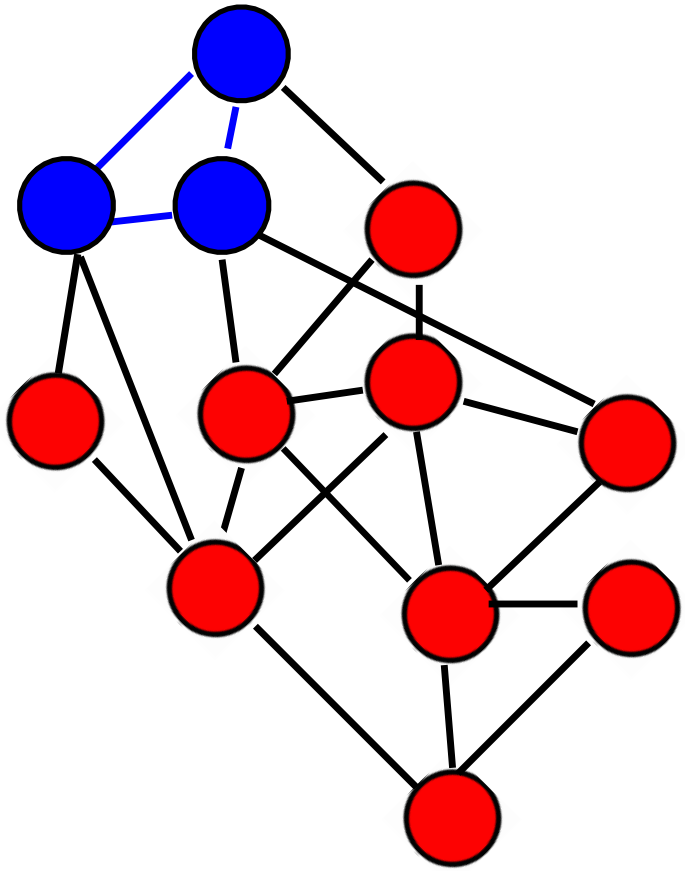
Full Replication Architecture

The Full Replication Architecture replicates the look up index across many servers.



Hash Architecture

The Hash Architecture makes the servers a more cohesive “unit”.



Structured dP2P Overlay Networks

Structured dP2P overlay networks are defined by 4 primary factors:

- a defined behavior for joining the network
- an overall defined naming scheme
- a routing structure and definition
- a defined behavior for leaving the network

Pastry

Nodes in the Pastry [1] network have:

- a numeric `nodeId`
- a state table that is comprised of:
  - a leaf set
  - a neighbor set
  - a routing table

Routing in Pastry

The routing scheme in Pastry is defined by the individual traffic being delivered.
In order for routing to occur:

- a destination is chosen (by key, or “name” of the node).
- the beginning node A chooses the node (B) that has the most similar `nodeId` to the destination node from A’s routing table.
- the message being sent to the destination node gets passed through B, and B will perform the same action.

This model ensures (due to the structure defined for Pastry), the message will reach the destination node.

Pastry Join Behavior

Pastry uses a specific metric for node proximity.
This metric is also used when a node attempts to join the network. If Node A attempts to join, it will use the proximity metric to find the nearest node to it (B).
Node B becomes the proxy through which the join process takes place:

- Node A routes a special “join” message to it’s own `nodeId` through Node B.
- The node furthest down the path to where Node A should be will then return it’s current state table.
- Node A the copies this state table, and modifies it so that it matches the specifics of Node A.

Pastry Leave Behavior

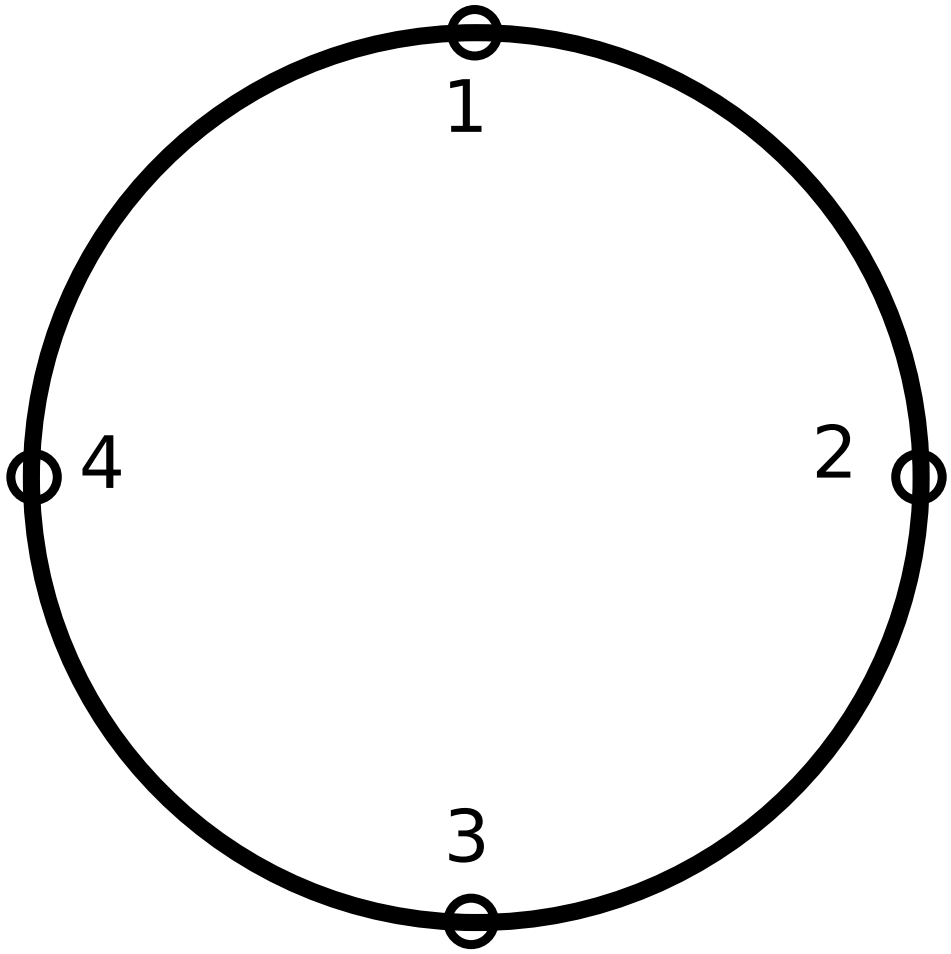
When a node (Z) attempts to leave the network, it will not be noticed until an attempt is made to route data through Z.
When a routing failure occurs:

- the node needing to pass along a message (node M) will try the next node (N) in the appropriate row in the routing table.
- M will then request the appropriate information from N regarding Z's location in the routing table, so as to account for the lost node.

M will try all of the nodes in the node that contained Z, or will move on to the next row if none of the nodes respond. This process may never complete, but this is statistically unlikely.

Chord

The nodes in Chord [2] are arranged in a ring, by name.



Join/Leave Behavior in Chord

- Nodes in Chord only have an understanding of the node immediately before and the node immediately after a given node in the ring.
- This means when a node joins, it becomes the new successor to the node immediately before, and the node immediately after becomes it's successor.
- Similarly, when a node leaves, only the two nodes immediately on either side of the node need to be notified, so that they may update their understanding of who their successor is.

Answering the question

Are distributed peer-to-peer overlay networks worth the effort?

Questions? Comments?


A. Rowstron and P. Druschel.
Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems.
In *Middleware 2001: IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg, Germany, November 12-16, 2001. Proceedings*, page 329, 2001.


I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan.
Chord: A scalable peer-to-peer lookup service for internet applications.
In *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 149–160, 2001.