14 Peer to Peer (p2p) Services and Overlay Networks

Main Points:

- What is an Overlay Network?
- Gnutella - free form searching.
- Distributed Hash Tables - object location

14.1 Overlay Networks

- The Internet is successful because the management is decentralised, yet connectivity is maintained
- Can we build applications that allow people to connect their machines, yet maintain control over their machines?
- Yes, by building an overlay network connecting instances of the application
- Examples include the web, file sharing, DoS protection.

14.1.1 A Generic Overlay Network

- Rather than directly routing messages to target, the overlay routes messages between the constituent machines
- The overlay network builds routing tables to meet application needs

14.2 Gnutella

- Used to share mp3 files, and much other copyrighted content.
- Simple protocol based on flooding and caching.
- Many different implementations interoperate.

14.2.1 Definitions

Servent The entities making up the Gnutella network, emphasising that clients are servers and vice versa.

Ping A message sent into the network. When a Servent receives the *Ping* it responds with a *Pong* to the sender.

Pong A message containing one or more servent addresses and some information about the data it is sharing.

Query A freetext description of the data asked for. A servent responds with a *QueryHit* if a match is found against its local data set.

QueryHit Provides enough information to acquire the data matching the query.

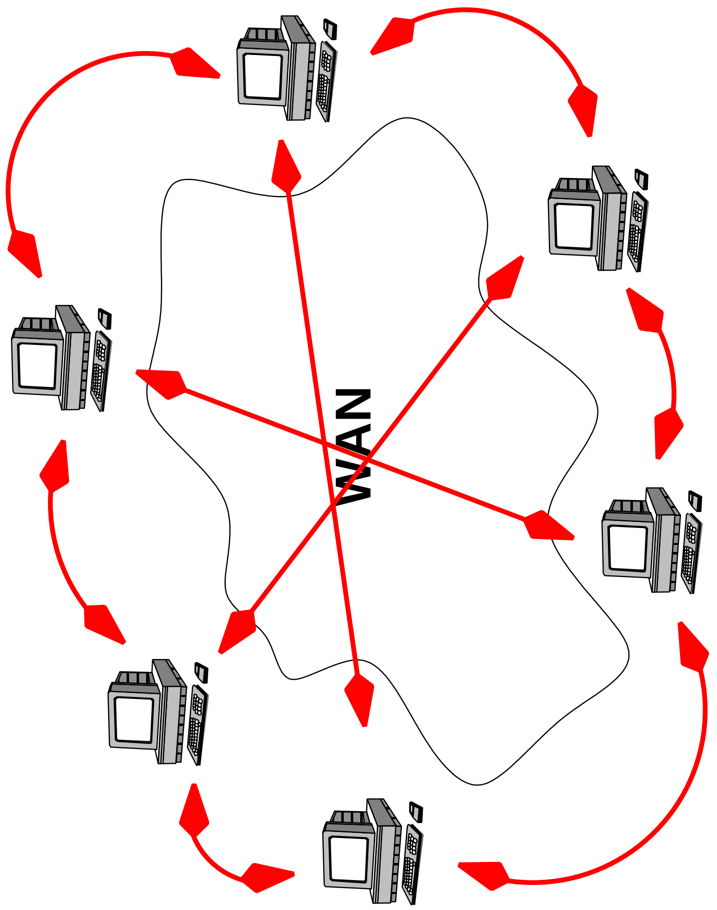

Figure 1: Generic Overlay Networks

14.2.2 Joining the Network

- A Servent joining the network can send a Ping request to discover other nodes.
- Servents receiving the Ping can choose to return a Pong with Servent details. It will then flood the Ping message to other Servents it is connected to.
- Pong messages return along the same path as Ping messages.
- Servents cache recently seen Ping identifiers so as to return Pong messages back along the same path. If the Pong identifier is not in the cache, then it will remove the Pong message.
- There is a TimeToLive field in each message decremented each time the message passes through a Servent. When the field reaches zero, the message is discarded.
- Servents can choose to initiate connections to other discovered Servents.

14.2.3 Queries

- Queries are flooded through the network in a similar manner to Pings.
- When a Servent receives a Query, it returns a QueryHit if the query text “matches” the search criteria

- How the Servent interprets the search field is entirely a local matter. It could only do exat matching, or just return everything in its dataset.
- The QueryHit returns along the same path as the Query, allowing caching.
- The QueryHit holds a Servent specific identifier and information about the file.
- If the user chooses to download the file, it initiates a direct connection to the holding Servent and retrieves the data using http on the Gnutella port.

14.2.4 Bootstrapping

- GWebCache is a set of web servers which store the IP addresses of "up" Gnutella hosts.
- Most clients now implement some form of UltraPeer, where machines which well-connected become better than others and have 10-100 leaf nodes and ¡10 connections to otehr ultrapeers.
- Ultrapeers can then filter traffic for leaf nodes using the uploaded tables from the leaf nodes.
- Ultrapeers are the nodes who advertise themselves in the GWebcache.

14.2.5 Gnutella Issues

Security How can you trust the data you've downloaded? Gnutella well-known for carrying viruses and trojans.

Bandwidth Usage When a node becomes an UltraPeer, the bandwidth usage increases enormously. This can cause problems for the institution in which the machine is sited.

FreeLoaders The network works because people share files and donate bandwidth and disk space. What happens when people just take without giving?

Copyright Most of the material on Gnutella is copyright to someone or other. Should people abuse copyright in this way?

14.3 Chord - A Distributed Hash Table Example

- When the name of the object is known, there are more efficient search structures, such as Hash Tables.
- A hash table takes an input key k, calculates the hash function on the key $h(k)$, and uses this as an index into a table.
- In a *distributed hash table*, the table into which $h(k)$ indexes is distributed across the nodes in the DHT.
- The key is provided, the hash function is calculated $h(k)$, and $h(k)$ is used to route to the node which would hold the object corresponding to the key.

14.3.1 Basic Operations

Chord is a research project which designed one of the first DHTs. Many others have now been designed.

Function	Description
`insert(key,value)`	Inserts a `key/value` binding in the DHT.
`lookup(key)`	Return the value associated with `key`.
`join(n)`	Causes a node to add itself into the Chord system.
`leave()`	Causes a node to leave the Chord system

Table 1: The Chord API

14.3.2 Identifiers and Keys

- A node generates its identifier by picking a value randomly from the hash space eg the 128 bits of SHA applied to its dns name.
- The node joins the DHT and determines who its predecessor and successor are in the table.

  Predecessor(n) The node with the highest identifier less than than n's identifier, allowing for wrapround..

  Successor(n) The node with the lowest identifier greater than n's identifier, allowing for wrapround.

- A node is then responsible for its own identifier and the identifiers between its identifier and its predecessor's identifier.

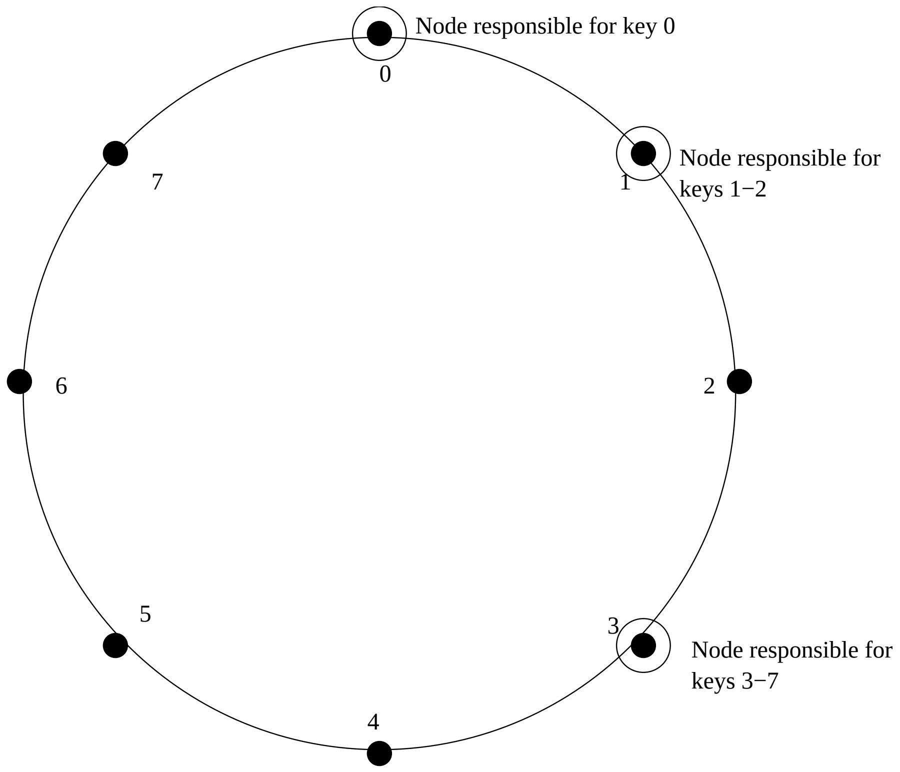
Node responsible for key 0
0
7
1
Node responsible for keys 1−2
6
2
5
3
Node responsible for keys 3−7
4

Figure 2: Identifier circle for 3 bit identifier

The Identifier Circle

14.3.3 Routing in Chord

Idea: Route to a node that halves the distance to the node responsible for the target key.

So for identifier length N,

- Each node with identifier n maintains a table with the address of nodes that succeed identifiers $n + 2^0 modN$, $n + 2^1 modN$,$n + 2^2 modN$ through to $2^{N-1} modN$.
- To route to a target k, look up the identifier in the routing table that precedes k, and pass the request onto this node.

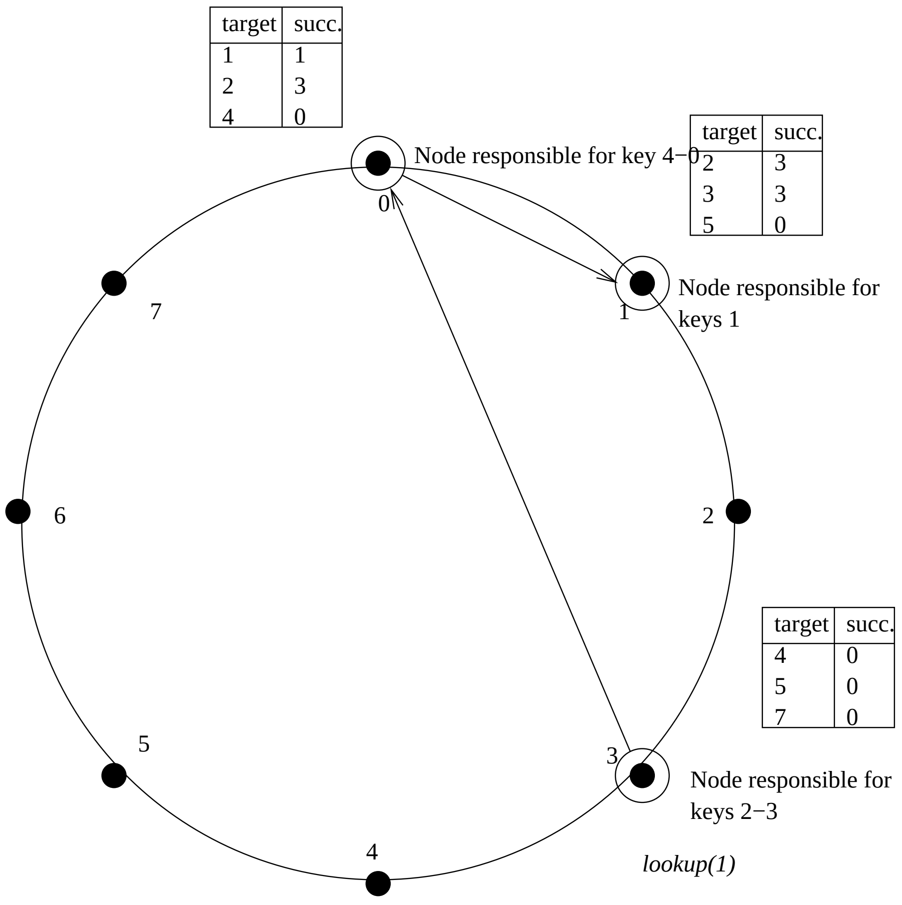
target succ.
1 1
2 3
4 0
Node responsible for key 4−0
target succ.
2 3
3 3
5 0
Node responsible for keys 1
target succ.
4 0
5 0
7 0
Node responsible for keys 2−3
0
1
2
3
4
5
6
7
lookup(1)

Figure 3: Routing example - *lookup(1)*, going from node 3 to the node responsible for 1

Routing Example www.pdos.lcs.mit.edu/chord/

Expected Number of Routing Hops

- Each hop in the routing will, on average, decrease the distance to the target by one half the current distance.
- If the size of the identifier space is N, then the average number of hops is, with high probablity, $O(log(N))$.

14.3.4 Maintenance: Joins and Leaves

On joining:

- The node generates its identifier n.
- A node locates its immediate successor by looking up n.
- The node then transfers the `(key,value)` pairs from its successor for identifiers in the DHT between n and the successor identifier.
- Update the routing tables of the relevant nodes
  ```
  for i = 1 to N
  p = findPredecessor($n - 2^{i-1}$)
  p.updateTables()
  ```

14.4 Current Research Challenges

- Gnutella is good for open search (as is Google)
- DHTs are good for object location
- Can we build overlay networks which provide looser search porperties, yet retain the $O(log(n))$ messages of DHTs?

14.5 Summary

- Overlay networks are an extension of the Internet design philosophy to distributed applications.
- They provide routing mechanisms that are robust to individual failures, and provide redundant paths.
- Genesis of much of the current research in networks and distributed systems.