Peer-to-Peer Networking

P2P Overlay Networks

Lecture content

- P2P network characteristics: graphs
- What are P2P overlay networks
  - Classifying P2P systems according to overlays
- Structured and unstructured systems
  - Can be made as a pure or hybrid P2P system
  - Characteristic from both main types
- P2P systems will be studied concerning the message routing/searching and implementation

Network Topologies

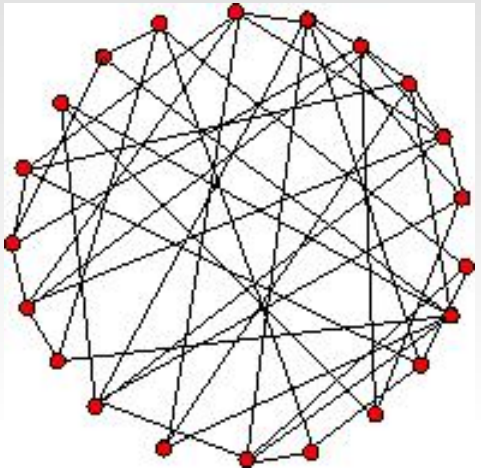
- Why studying network topologies?
  - Important aspect for the performance and resilience properties of networks
  - Improving routing in Gnutella: analytical topology models could be created and used in network simulators (Consider setting up a 40 000 nodes test environment?)
- Mathematical laws concerning distributions and graph theories can be applied to network research

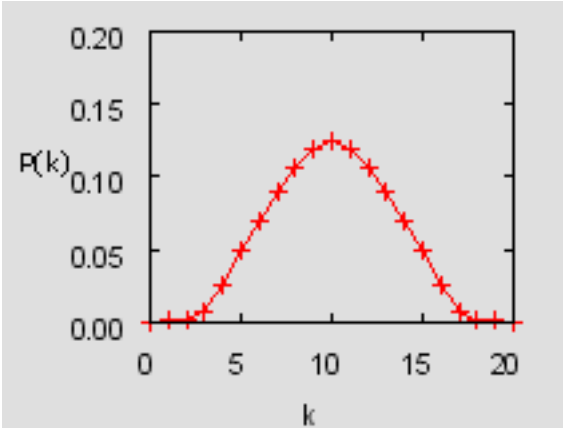
Graphs

- Network topology can be represented as a graph composed of:
  - A set of nodes or vertices to represent the entities or agents that communicate using the network
  - A set of edges or links that represent the communication channels that connect the nodes
    - Communication channels can allow information to flow in only one or both directions
- Degree is the number of links that a node has

Random Graphs

- First introduced by Erdös and Rényi
- Graph *G* constructed from model G(n,p): *n* nodes, for each possible edge, add the edge to *G* according to probability *p*
- Number of edges per node follows normal distribution; nodes tend to have approximately the same amount of edges
- Random graph theory well known, but real networks seldom are random made

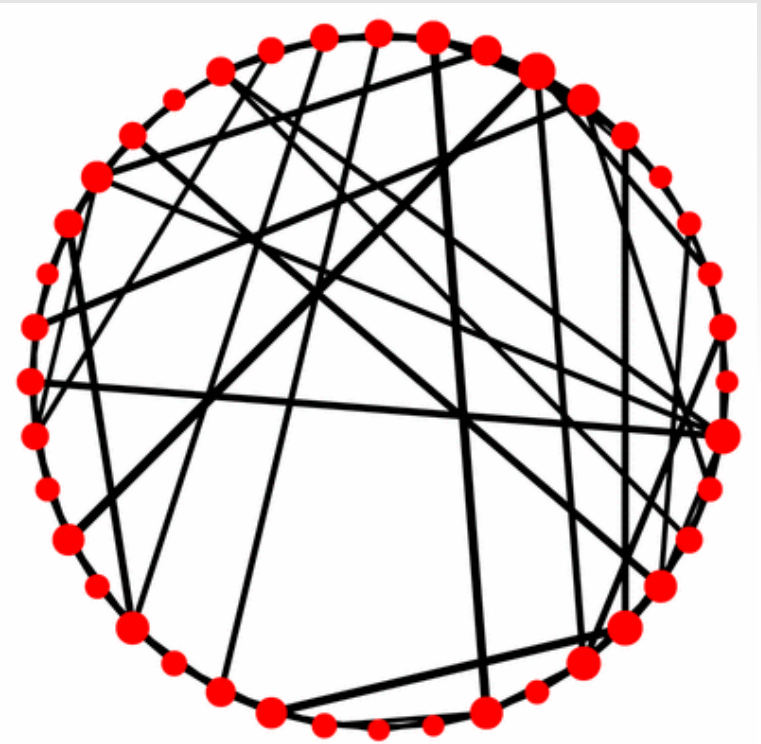




[https://nwb.slis.indiana.edu/community/?n=ModelData.RandomGraph]

Small World Networks

- Watts and Strogatz: randomly rewire a regular graph
- Barabási and Albert: Links not added randomly to the network nodes but rather in proportion to the number of links that the nodes already have
- These two models create graphs which are often termed *small world networks* (SWN)
- Stanley Milgram observed similar behaviour in social networks already in 1967
  - Six Degrees of Separation



[http://www.math.cornell.edu/~durrett/RGD/pix.html]

Six Degrees of Separation

- People's ability to find routes to a destination within the social network of the American population
  - Several packages to random people, asking them to forward the package, by hand, to someone specific
  - If receiver did not know the target directly, they should sent it to someone known who might be closer
- He concluded that people were remarkably efficient at finding such routes, even towards a destination on the other side of the country

…Back to Small World Network

- SWNs have comparatively small average path length between two arbitrary nodes
- Majority of nodes has relatively few local connections to other nodes
- But a significant small number of nodes have large wide-ranging sets of connections
- Enables efficient short paths because these well-connected nodes provide *shortcuts*

SWN Pros and Cons

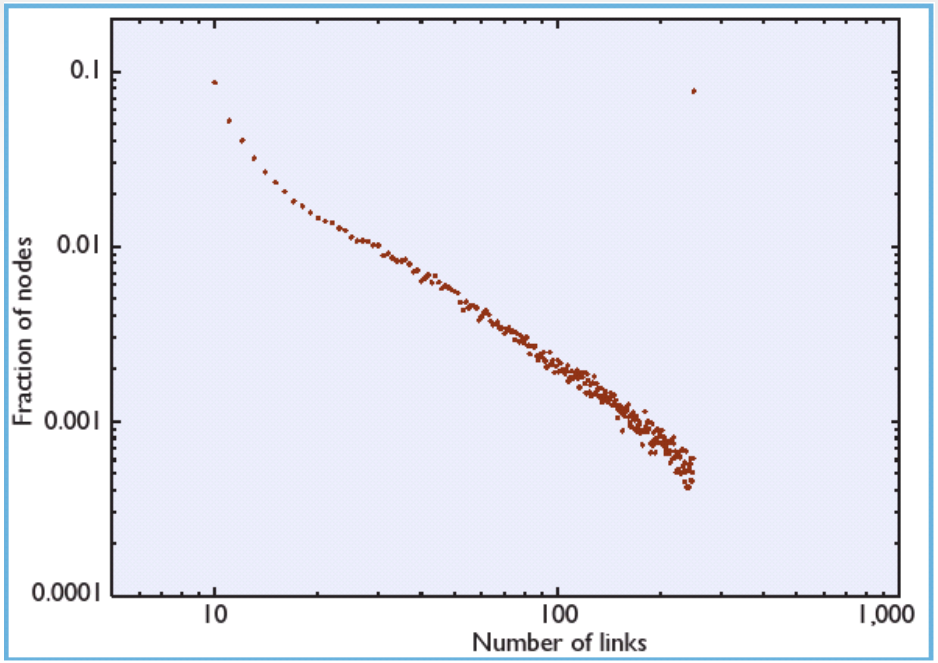
- Highly resilient random to node failures and attacks
  - Even after removing 80% of the nodes, the network can stay connected
  - The more connected “hubs” keep it together
- On the other hand, SWNs are highly vulnerable to coordinated attacks against these well-connected hubs
  - Separates network into non-communicating segments

Can P2P Network Evolve to a Small World Network?

- Answer is yes: through *preferential attachment* defined in Barabási-Albert model
  - Means that the more connected a node is, the more likely it is to receive new links
  - Gnutella: Nodes linked to many other nodes spread the knowledge about their existence more efficiently thus having higher probability connecting the new nodes
  - Freenet: Arrival of a new node is propagated randomly according to the routing table in each peers; more linked nodes have higher probability to be linked to new nodes

Power Law Distribution of Graph Degree

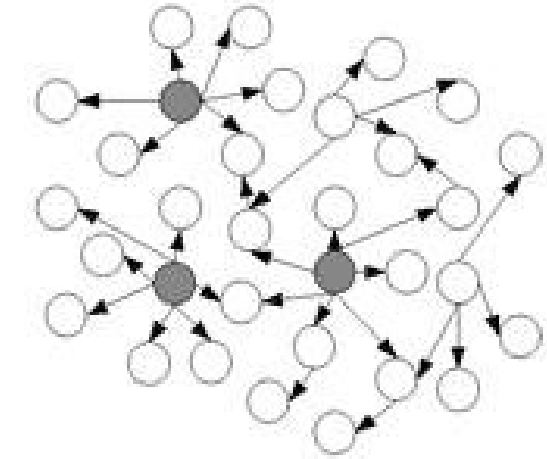
- Power law distribution of graph degree tends to arise naturally when networks grow by preferential attachment
- Many networks, including the Internet and P2P networks
- Figure presents degree distribution among Freenet nodes
- The network shows a close fit to a power-law distribution
- Log-log scale
- General form: $p(x) \sim x^{-t}$, here $t = 1.5$



[Clarke et al., Protecting Free Expression Online with Freenet]

Scale-Free Networks

- Power law distribution of graph degree is a defining characteristic for scale-free networks
- The network is held together by a few highly connected hubs
- Scale-free network properties are independent of the number of nodes
- The scale-free network's topology is a natural result of the expanding nature of real networks



[http://www.macs.hw.ac.uk/~pdw/topology/ScaleFree.html]

Graph Theory Summary

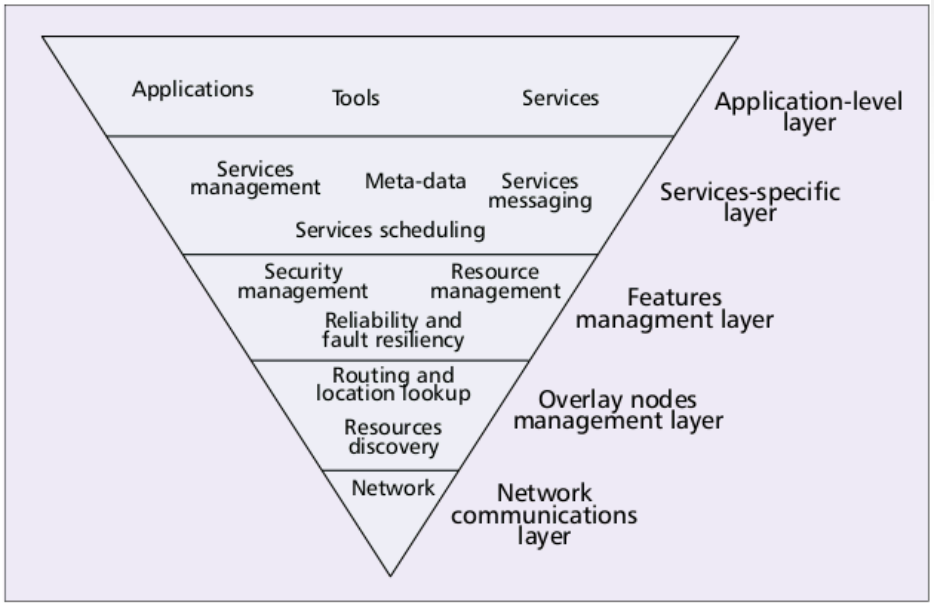
- P2P networks often follow the small-world network model
- Benefits: good scalability and fault-tolerance
- Weaknesses: vulnerable to targeted attacks
- Also known in theoretical level; can be used to construct topology models for testing and developing
- Helps to understand behaviour in large networks

Overlay Networks

- P2P systems are implemented as virtual networks of nodes and logical links on top of an existing network
- Typically, on top of the Internet
- These virtual networks are called P2P overlay networks or P2P routing substrates
- Self-organising and distributed systems in nature
- Overlays are one way to classify the vast amount of different P2P designs and architectures

P2P Abstract Overlay Architecture

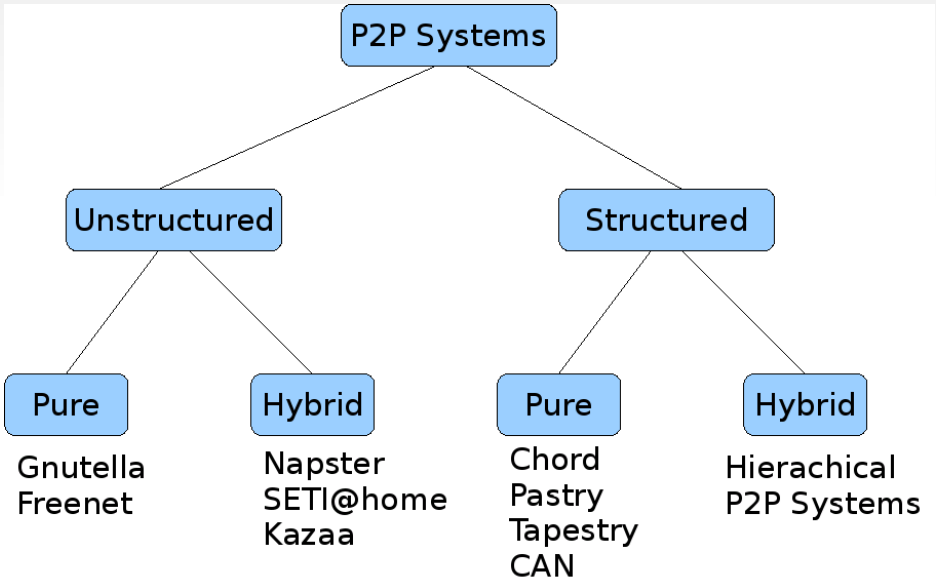
- Network characteristics of a desktop machines connected over Internet
- Management of peers: routing and discovery
- Feature management: security, reliability, aggregated resources
- Services-specific: scheduling tasks, meta-data describing stored content and location information
- Application-level: tools and services with specific capabilities



[Lua et al. A survey and comparison of peer-to-peer overlay network schemes]

Taxonomy of P2P Overlays

- Simply classification according to the overlay topology
- In structured systems, the overlay is strictly controlled by the system
- In unstructured, there are no explicit control how network is arranged
- In pure P2P all nodes are equal
- In hybrid P2P subset of nodes have special roles; centralised index or super-peer layer
- Structured hybrid is yet rarely implemented



Taxonomy Notes

- Taxonomy presented above is only a one way to classify or name different systems and design architectures (according to the overlay network)
  - There are numerous different models, more is coming and current designs are being merged...
- In here, hybrid systems cover both centralised and hierarchical super-node models
- P2P is still relatively new research field and is subject to fast development

P2P Taxonomy Overview

- First generation of P2P systems were unstructured
  - Central index directory in Napster
  - Flooding queries in Gnutella
  - Later super peers in Gnutella2 and Kazaa
  - Limited scalability and single point failure due to centralising and flooding searching inefficiency
- Second generation designed to solve these issues by using a structured network, in which nodes are self-organising and decentralising the system

P2P Taxonomy Overview

- In structured network nodes (peers) are, at least partially, aware of the topology and neighbours
- Messages can be routed deterministically by a basic key-based routing mechanism
  - Peers and data objects have identifiers (keys)
  - Routing message (query) for an identifier (key) towards closer node that is holding the data
- Examples: Tapestry, Chord, Pastry, and CAN

Unstructured Overlays

- Peers organised in a random graph
  - Flat, pure P2P
  - Hierarchical via super-peers
- Routing messages
  - Flooding
  - Random walk
  - Expanding-ring Time-to-Live
- Peers evaluate queries locally on their own content

Unstructured P2P Systems

- Unstructured pure P2P
  - Freenet
  - Gnutella
  - BitTorrent
  - Overnet/eDonkey2000
- Unstructured hybrid P2P
  - Napster
  - FastTrack/Kazaa
  - SETI@home

Pure Unstructured Overlays

- No precise control over the network topology
- Nodes joining the network by some loose rules
- Extremely resilient to nodes entering and leaving
- File placement not based on any knowledge of topology
  - Nodes have to query neighbours for data
  - Search mechanisms can be extremely unscalable
- Complex queries supported (also in hybrid model)

Unstructured Hybrid Overlays

- Hybrid of a client-server model and a pure P2P model
  - Locating resources: Napster central directory index
  - Coordinating resources: SETI@home
- Hybrid model includes also the hierarchical unstructured network
  - Super-peers provide an upper layer
  - Gnutella 1.0 proposal and FastTrack/Kazaa
- See previous lecture for more design details

Structured Overlays

- Network topology tightly controlled by P2P system
- Content placed at specified locations, not random peers
- Efficient query routing inside the structure
- Scalability guarantees on numbers of hops to answer a query: major difference to unstructured overlay
- Based on the Distributed Hash Table (DHT)

Distributed Hash Table

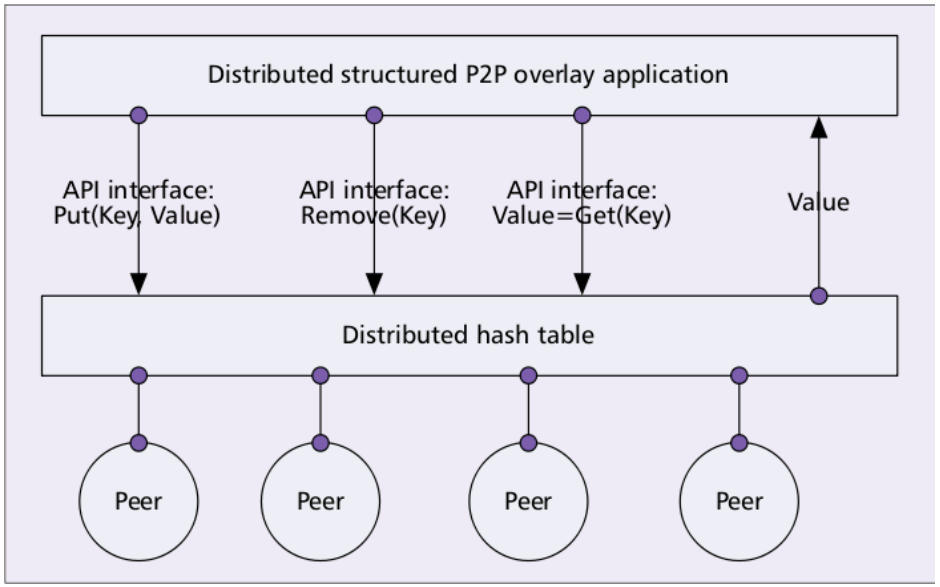
- DHT created for improving the search scalability issues of P2P
  - Centralised: bottleneck, single point of failure
  - Flooding (unstructured, Gnutella) or hybrid search: cannot guarantee discovery, poor on rare items
- Identifiers mapped uniformly to all nodes (i.e. peers)
- Data objects assigned unique identifiers *(keys)* from the same identifier space

Distributed Hash Table

- Core operation: find node responsible for a key
  - keys mapped to nodes
  - Efficient routing of request to this node (put/get/remove)
- Generic DHT
  - Node ID: *m*-bit identifier for node (“IP address”)
  - Key ID: *m*-bit identifier for data item (“file name”)
  - Value: sequence of bytes (“file content or reference to the content”)

Generic DHT interface

- Put(key, value): store {key, value} pair at the node responsible for that key
- Value = Value(Key): retrieve value associated to the key from the appropriate node
- Remove(Key): Remove {key, value} pair at the node responsible for the key



[Lua et al. A survey and comparison of peer-to-peer overlay network schemes]

DHT Principles

- Each peer maintains a small routing table consisting of its neighbouring peer IDs and their IP addresses
- Lookup queries/message routing forwarded across overlay paths to the *closer* identifier in the space
- In theory, can guarantee finding any data object in O(*log N*) overlay hops, *N* is the number of peers
- Underlying network path can significantly differ from the overlay path: high latencies

DHT Problems

- Theoretical performance good, but in practice?
  - High latencies: physical topology not corresponding to the logical layout
  - Higher overhead than unstructured P2P networks for popular content (e.g. flooding for highly replicated data)
- No support for complex queries
- Data coherence (store a copy/pointer to data objects)
- Assumes all peers equally participating: bottle-neck at low capacity peers

DHT Applications

- File sharing, rendezvous based communication (BitTorrent-alike)
- Chat service (Circle)
- Web caching (Codeen)
- Databases
- Naming services (DNS replacement)
- Publish/Subscribe

DHT Protocols and Implementations

- DHT is a generic interface, several implementations
  - CAN (Content Addressable Network)
  - Chord
  - Kademlia
  - Pastry
  - P-Grid
  - Tapestry
  - Viceroy

Chord

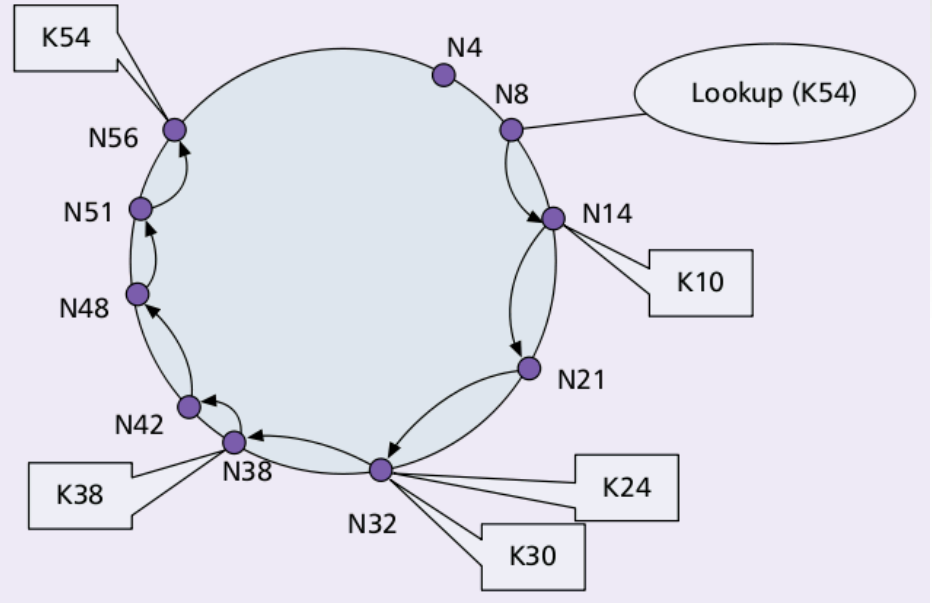
- Consistent *SHA-1* hashing to assign identifiers
  - m-bit identifier (e.g. 160-bit) space for node/object
  - Even distribution of nodes and keys on the overlay
- Nodes and objects ordered on an *identifier circle* indexed from 0 to 2^m-1 (modulo 2m)
- Node is responsible for objects (keys) between its predecessor and itself
- Response to a query is sent via reverse overlay path

Chord

- Modifying the ring causes successor pointer update
  - Correctness relies peers being aware of its successors
- A stabilisation protocol in background periodically
- Robustness: maintain successor list size of *r* to avoid failing peers cutting the ring
- Applications
  - Co-operative mirroring or co-operative file system (CFS)
  - Chord-based DNS (independence from root servers)

Chord Ring: Simple Key Location

- N8: lookup in finger table the furthest node that precedes key (K54)
- Simple key location using successor nodes

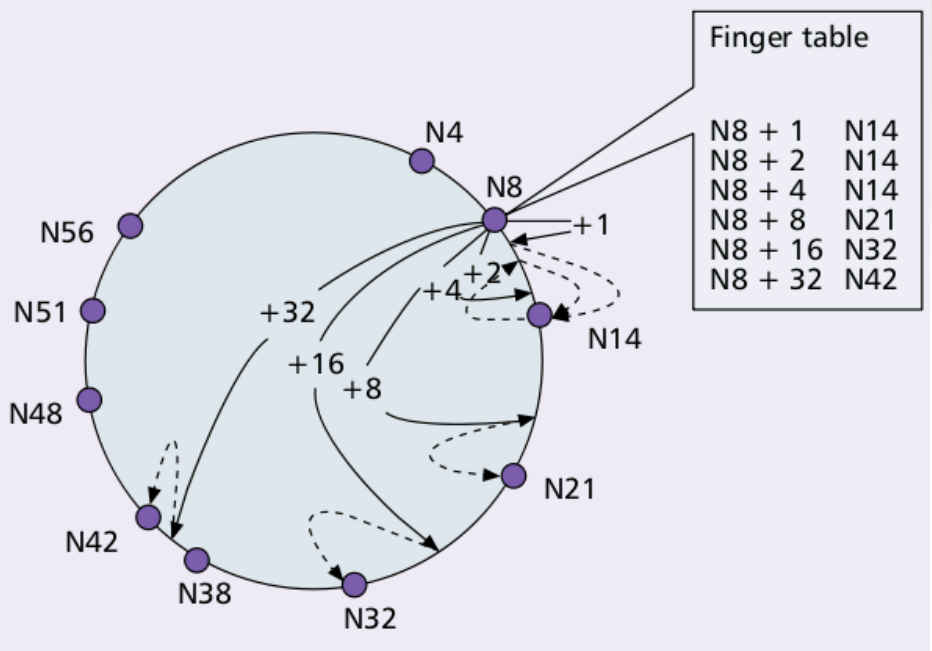


[Lua et al. A survey and comparison of peer-to-peer overlay network schemes]

```
// ask n to find the successor of id
n.find_successor(id)
      if ( id ∈ (n, successor) )
            return successor;
      else
            // forward the query around the circle
            return successor.find_successor(id);
```

Chord Ring and Finger Table

- Finger table for maintaining routes of neighbours
- *m*-bit (here $m = 6$) node keys are arranged in a circle
- *N* is the number of peers
- Node has route data for *m* neighbours, kept in the finger table
- The i^{th} entry in the table at peer *n* contains the identity of the first peer *s* that succeeds *n* by at least $2^i - 1$, i.e., $s = \text{successor}(n + 2^i - 1)$, when $1 \leq i \leq m$



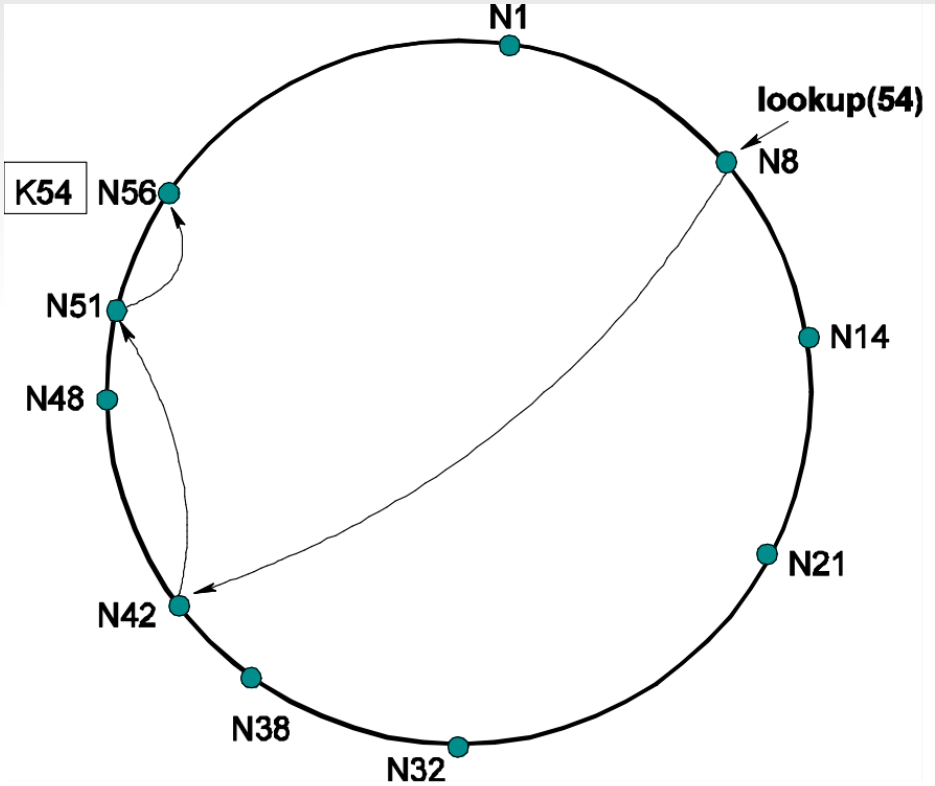
[Lua et al. A survey and comparison of peer-to-peer overlay network schemes]

Chord Ring: Scalable Key Location

- Lookup the furthest node in finger table that precedes key

```
//ask n to find the successor of id
n.find_successor(id)
      if ( id ϵ (n, successor) )
            return successor;
      else
            //forward the query around the circle
            n0 = closest preceding node(id);
            return n0.find_successor(id);

//search the local table for the highest
//predecessor of id
 n.closest_preceding_node(id)
      for i = m downto 1
            if( finger[i] ϵ (n, id) )
                  return finger[i];
      return n;
```



[Stoica et al. Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications]

Chord Performance

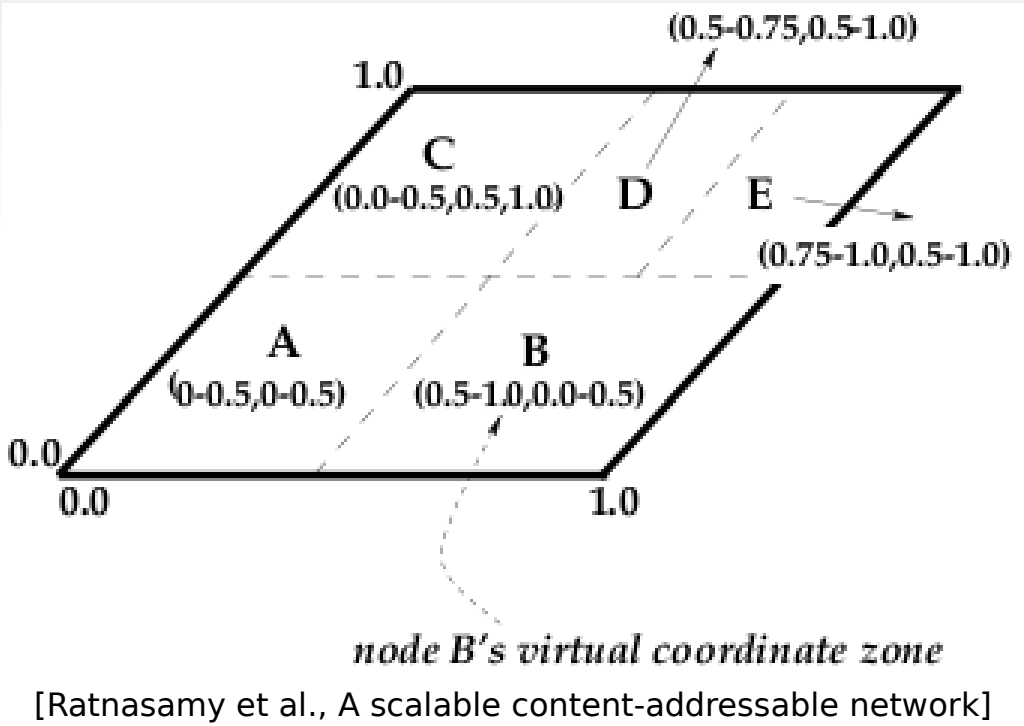
- Simple key location routes half-way across the ring on average: $O(N)$ hops required (how bad is it?)
- Scalable version: $O(\log N)$ hops
  - Using finger tables, the distance could be nearly halved at each step
- Small overhead for finger table upkeep, joining and entering peers add $O(\log^2 N)$ overhead
- Logical topology differs from physical causing long hops and greats latencies

Content Addressable Network (CAN)

- Hash table functionality on an Internet-like scale
- Architectural design: “virtual multi-dimensional Cartesian coordinate space on a multitorus”
- Coordinate space dynamically partitioned among peers, each peer owns a distinct zone in the space
- peer maintains a routing table that holds the IP address and virtual coordinate zone of each of its neighbour coordinates

Coordinate Overlay

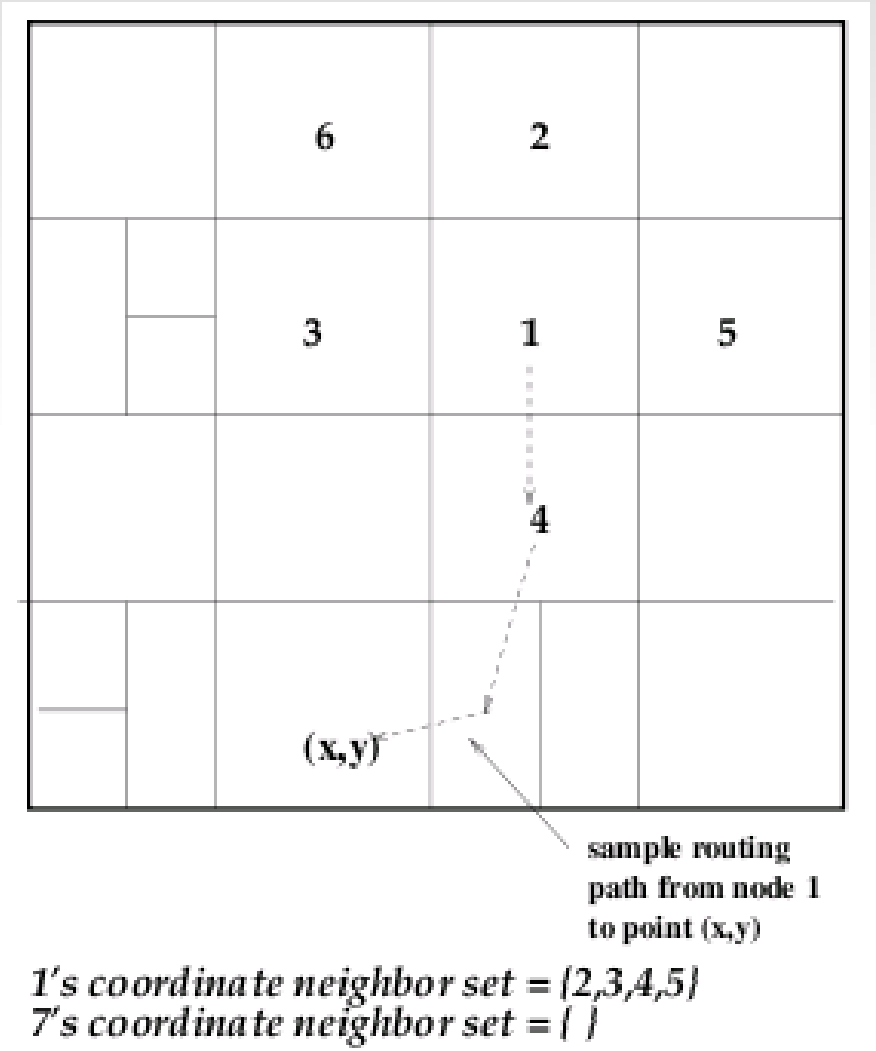
- Example of 2-d space with 5 nodes
- Overlay actually quite simple to understand
- Hashes calculated one per each dimension
- Torus wraps around the values
- E.g. node C holds data objects which are are hashed x ={0.0-0.5}, y = {0.5-10.0}



[Ratnasamy et al., A scalable content-addressable network]

Routing in CAN

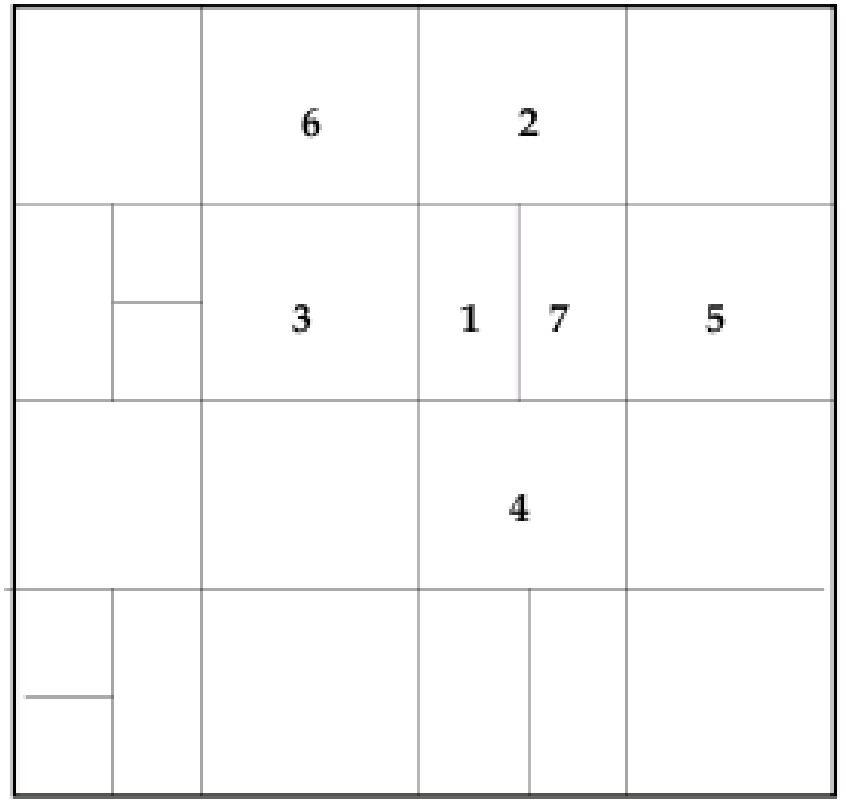
- Example: node 1 sends message to point (x,y)
- Each key hashed to a *point* (node) in the space
- Simply greedy forwarding to route message to the neighbour with coordinates closest to the destination coordinates
- Data objects stored at points



[Ratnasamy et al., A scalable content-addressable network]

Node Arrival in CAN

- Example: node *7* arrives to CAN
- New node must find a node already in CAN: get its IP address by some mechanism, e.g. bootstrapping
- Find a node whose zone will be split (e.g. random coordinates)
- Neighbours of the split zone must be notified to include the new node to routing



1's coordinate neighbor set = {2,3,4,7}
7's coordinate neighbor set = {1,2,4,5}

[Ratnasamy et al., A scalable content-addressable network]

CAN Performance

- N nodes, d dimensions
- Routing complexity: $O(d * N^{1/d})$ hops
- Node state complexity: $O(d)$
- Improvements to CAN
  - More dimensions improve routing
  - Multiple realities: maintain multiple, independent coordinate spaces with each node for availability and fault tolerance

CAN Usage

- In practise, haven't been used so much
- Offers efficient insert and retrieval of content in large distributed storage area network with a scalable index mechanism
- Suitable applications include
  - large-scale storage management (such as OceanStore)
  - Wide-area name resolution services

Plaxton Mesh

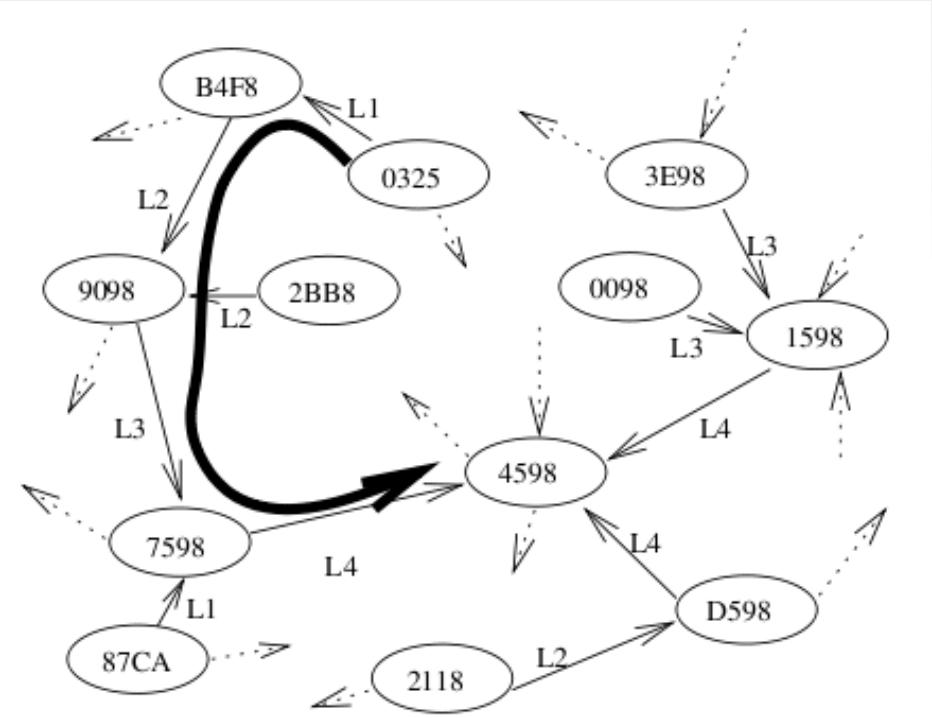
- Prefix/postfix routing, used in Tapestry and Pastry
  - Similar to CIDR IP address allocation
  - E.g. ***7 --> **97 --> *297 --> 3297
  - Node ID and object ID hexadecimal, Base 16 (40 digits)
- Peer's local routing map has multiple levels
  - Each level represents a match of the suffix with a digit position in the ID space
  - The i^{th} entry in the j^{th} level is the ID and location of the closest node which ends in "i"+suffix(N, j-1)

Plaxton Mesh

- Means, that level 1 has links to nodes that have nothing in common, level 2 has the 1st digit, etc.
- For example, the 9th entry of the 4th level for node 325AE is the node closest to 325AE in network distance which ends in 95AE (postfix resolving)
- Routing takes $\log_B N$ hops, where B = 16, N = peers
- Closest node also closest in IP sense (RTT)
- In Pastry the identifiers arranged in a circle (Chord)

Plaxton routing in Tapestry

- Message from 0325 to 4598
- Start: B4F8 matches ***8
- Next router: examine the (n + 1)th level map to locate the entry matching the value of the next digit in the destination ID
- The n^{th} peer that a message reaches shares a suffix of at least length n with the destination ID
- Only part of nodes and links shown



[Zao et al. Tapestry: an Infrastructure for Fault-Tolerant Wide-Area Location and Routing]

Structured Hybrid Overlays

- Idea of using super peers to reduce maintenance cost in Chord and others DHT networks
  - DHT-based systems often praised for its guaranteed search feature but has relatively higher maintenance overhead than Gnutella-like unstructured P2P networks
- Some studies available
  - Joung, Y. and Wang, J. 2007. Chord2: A two-layer Chord for reducing maintenance overhead via heterogeneity. Comput. Networks 51, 3 (Feb. 2007), 712-731. DOI= http://dx.doi.org/10.1016/j.comnet.2006.05.010
  - Zhu, Y., Wang, H., and Hu, Y. Super-peer Based Lookup in Structured Peer-to-Peer Systems. In Proceedings of 16th International Conference on Parallel and Distributed Computing Systems, Nevada, August, 2003.

Summary

- Basic taxonomy according to overlay networks
- Unstructured (pure/hybrid):
  - No explicit control over logical network topology: how to route effective? Flooding, random walk, heuristics, ...
  - Hybrid with super-peers or other specialised service
- Structured (mostly pure):
  - Overlay determines how the peers organise
  - DHT-based: Data discovery determined and effective

Summary

- Overlays are concerned about how the logical topology is mapped to the physical infrastructure
  - Great effect on data discovery and routing costs
- Ultimately, the best suited P2P overlay network depends on:
  - the applications and its required functionalities
  - Performance metrics: scalability, network routing, location service, file sharing, content distribution, etc.

Discussion

- Unstructured P2P overlay network
  - Centralised: cannot scale, single point failure
  - Flooding-request model: effective in locating popular data objects, but causes excessive network loads and cannot guarantee finding remote or rare data objects
- DHT-based systems are more efficient and offer strong theoretical fundamentals
- Still DHT is not suitable for mass-market file sharing, but why?

Discussion

- DHT does not capture the semantic object relationships between its name and its content or metadata
- Ability to find exceedingly rare items not essential for mass-market file sharing
- Efficient key-word search has not been proven yet
- Algorithms based on precise placement, but ability to handle unreliable peers is still under study

Discussion

- Freenet differs from the basic unstructured P2P
  - Indexing scheme based on content-hash keys
  - Provides anonymity and data integrity
  - Scalability and fault tolerance
- Other improvements have emerged
  - Clustering key space instead of LRU (for Freenet)
  - BitTorrent: download distribution protocol
- Super-peers in structured DHT-based systems?