Peer-to-Peer Overlays

Dr. Laurent Mathy

Computing Department

Lancaster University, UK

laurent@comp.lancs.ac.uk

LAAS, Nov 29, 2006

Aims of the talk

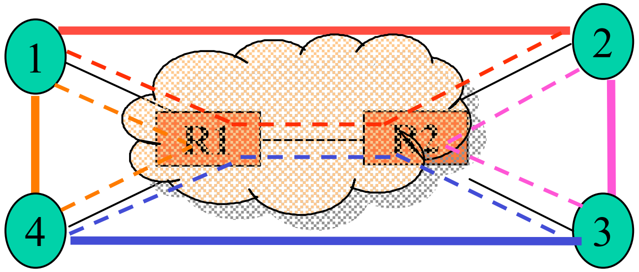
- To provide a high level introduction to P2P system design and their underlying generic mechanisms
  - Not the nitty-gritty of specific protocols
- Present the two broad classes of P2P systems
- Briefly outline possible applications of P2P systems
- Briefly outline security issues in P2P systems

Outline

- Definition
- Unstructured P2P systems
- Structured P2P systems
- Some applications
- Churn
- (Some) Security issues
- Conclusions

Definition

- **Overlay Network: virtual communications structures that are logically 'laid over' a “physical” network such as the Internet**
  - **Virtual/logical links: tunnels, application level “associations” (TCP, UDP), etc.**
    - **The overlay links are not “physically fixed”, they are “configured”**
    - **Underlay responsible for implementing the overlay links**
  - **Must get to know one overlay node (out-of-band) prior to joining**
- **P2P Overlay**
  - **Application-level overlay**
  - **(near) equivalent functionality on each node**
  - **Self-organisation**
  - **Geared towards object location/retrieval**

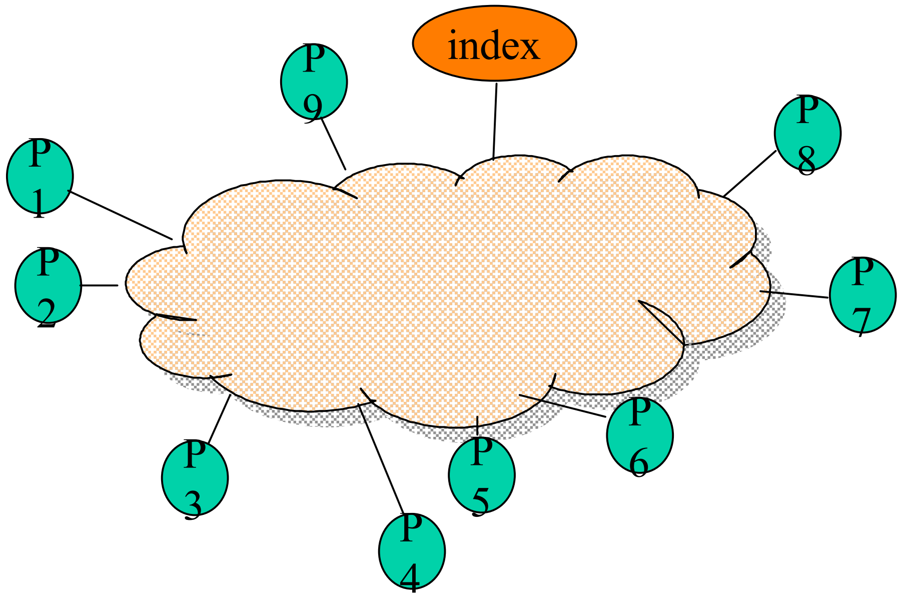


Outline

- Definition
- Unstructured P2P systems
- Structured P2P systems
- Some applications
- Churn
- (Some) Security issues
- Conclusions

Unstructured P2P

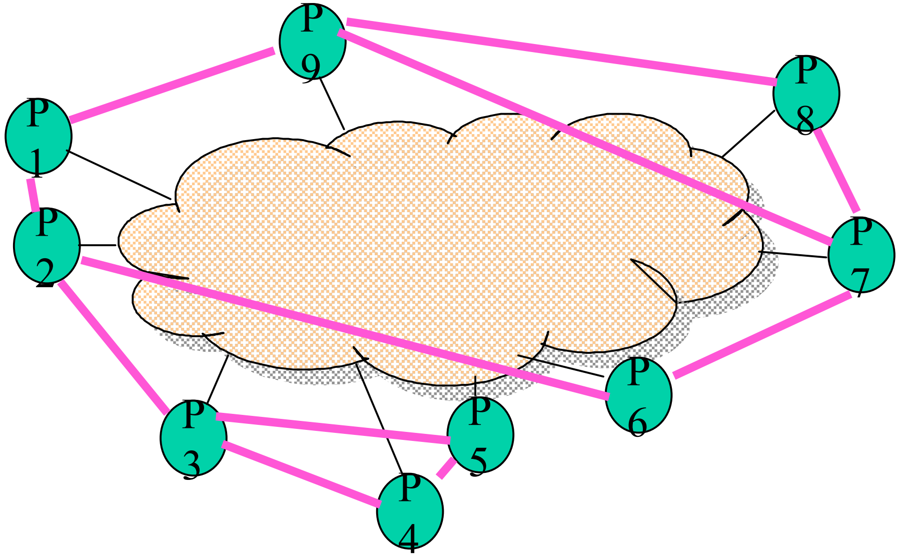
- The main issue in P2P is indexing/locating objects
  - Direct exchange between nodes once located
- First generation unstructured P2P system were based on central index
  - Register objects with index
  - Query index
  - Choose a peer as server
- Index is single point of failure



- Napster
- Bit-torrent
  - Object is block (block of file)
  - One index per file
  - Multi-source download

Unstructured P2P (2)

- Decentralised
  - Each node has its objects/references to objects
- Network structure is based on group members (i.e. you choose who to talked to based on who they are solely)
  - Builds a mesh-based structure where each node selects and tracks a few neighbours
  - Adaptable , “free” topology
  - Topology usually constructed in order to optimise some objective
    - low delays
    - Simplicity (random)



Unstructured P2P

- Search in decentralised unstructured P2P
  - Flood request on mesh (broadcast)
- All this flooding poses some scalability issues
- To improve scalability, trade accuracy for reduced traffic based on
  - Probabilistic techniques: Random walks, etc
  - Limited scope broadcast
- No guarantee that a search yields results

Outline

- Definition
- Unstructured P2P systems
- Structured P2P systems
- Some applications
- Churn
- (Some) Security issues
- Conclusions

Structured P2P

- Goal is to ensure scalability without compromising on false negatives
  - Other important goal is to try and provide a maximum bound on time to location of an object

- Network structure based on information structure (i.e. you choose who to talk to based on what piece of info you are after) as well as networks configuration.

- Here the focal point is the objects

Structured P2P

- Basic principle is the same regardless of system
  - Define an appropriate address space
  - Give nodes and objects an address
  - Split space between nodes
    - Each node is responsible for managing part of the space (region)
    - Nodes are responsible for the objects whose address falls into their region

- The different structured P2P systems are simply about
  - Allocating/re-allocating regions to nodes
  - Placing objects in the appropriate region
  - Efficiently locating objects in the space (finding node responsible for the object)

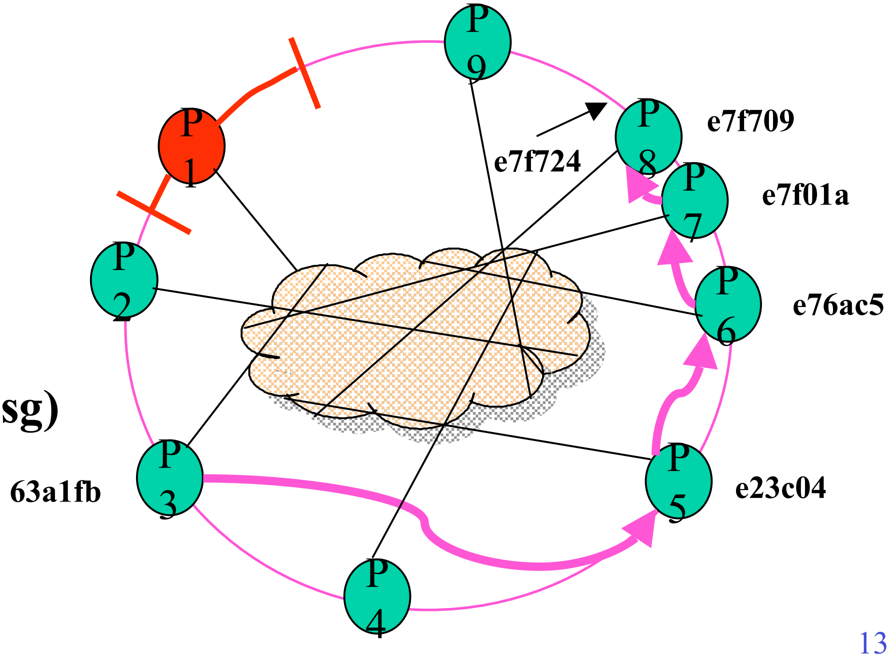
Structured P2P (2)

- "Appropriate address space"
  - Uniform address space
  - Roughly, the number of objects managed by each node should be, on average, the same
- Original structure of object representation (file name, keyword, URL, etc) must be abstracted
  - Use of (possibly multiple) hash functions to transform object representation into uniform address space
  - This representation of objects is often called a "key"
- Hence the name "Distributed Hash Table"
  - The structure manages (key, value) pairs

Structured P2P -- Pastry

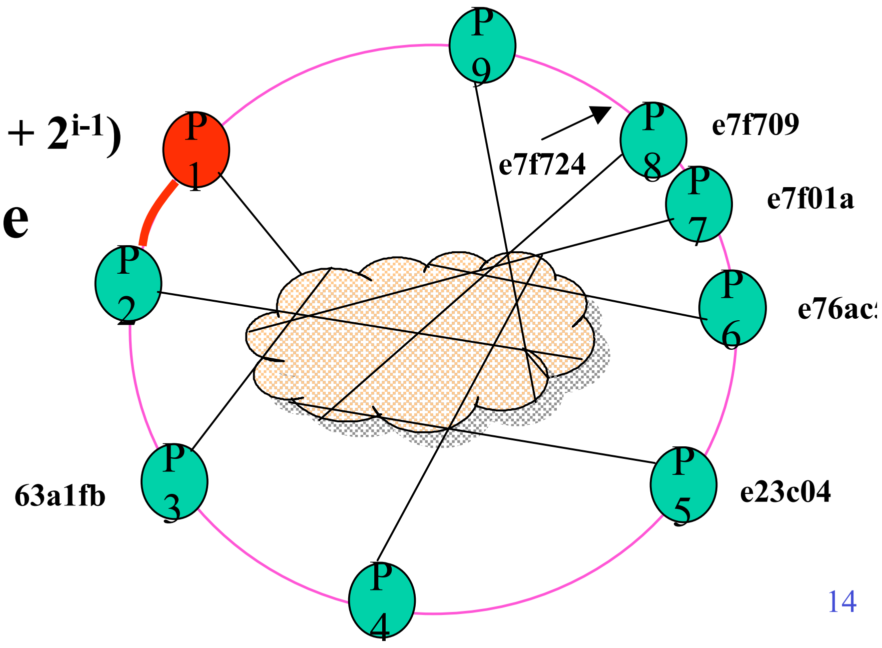
- Address space is a circle (often with 2^{128} addresses)
- Nodes and objects get Id on circle
- Space split: each node responsible for keys that are numerically closest to it
- Routing principle: forward requests through series of nodes known to have longer prefix-match with key than current node
- State per node: $O(2^b \log_{2b} N)$
- Object location: $O(\log_{2b} N)$

Route (e7f724, msg)



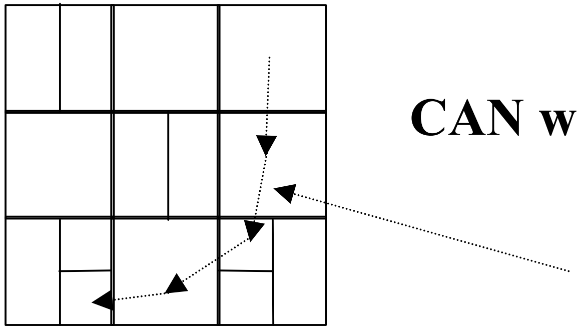
Structured P2P -- Chord

- Address space is a circle (often with 2^{128} addresses)
- Nodes and objects get Id on circle
- Space split: key assigned to first node whose Id is equal to or follows the key (concept of successor node)
- Routing principle: suffice to know the next successor node on circle to guarantee correct routing
  - But can be inefficient
  - Idea of “finger table”
    - Know successor nodes for keys (Nid + 2^{i-1})
- Object location and state per node
  - O(logN)



Structured P2P -- CAN

- Content Addressable Networks
  - A CAN is a virtual d-dimensional Cartesian coordinate space on a d-torus
  - Nodes have coordinates in the space and the space is partitioned in as many “zones” as there are nodes – each node “own” a zone
  - Content is “hashed” onto a coordinate
    - Coresponding zone owner holds either content or reference to it
  - Can is capable of routing message to a coordinate (actually owner of zone that contains the coordinate), in a hop-by-hop manner (i.e. From neighbouring zone to neighbouring zone)
- State: O(2d)
- Object location: $O(d/4N^{1/d})$



CAN with 15 nodes

Outline

- Definition
- Unstructured P2P systems
- Structured P2P systems
- Some applications
- Churn
- (Some) Security issues
- Conclusions

Some Applications

- **P2P systems provide an application substrate**
- **Possible applications**
  - **File sharing**
    - **Objects are either complete files or file chunks**
  - **General indexing (structured)**
  - **Storage**
    - **Split file into chunks, add redundancy (erasure coding) and store chunks on responsible nodes (possibly with redundancy)**
  - **Server selection (structured mostly)**
  - **Large-scale combinatorial search (structured)**

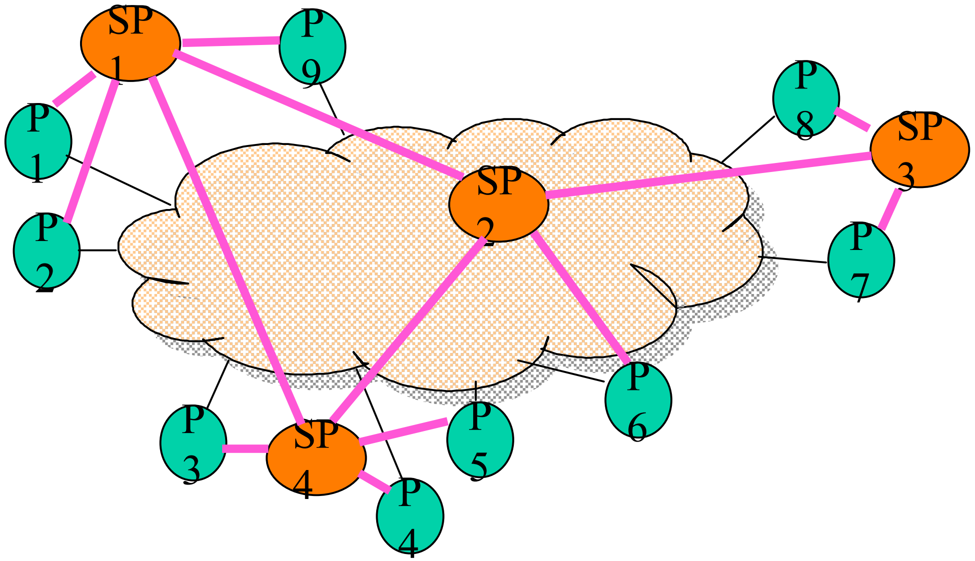
Outline

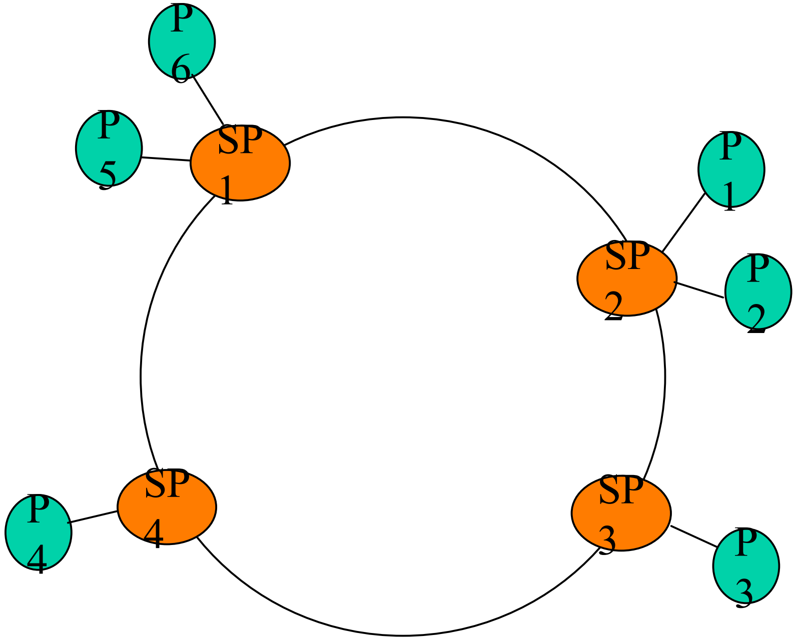
- Definition
- Unstructured P2P systems
- Structured P2P systems
- Some applications
- Churn
- (Some) Security issues
- Conclusions

Churn

- Churn represents the notion of nodes joining and leaving the P2P system
  - Churn creates much work for the P2P system
  - During churn
    - Some objects may be temporarily unlocatable
    - Some objects may be temporarily unavailable
    - Much of the protocols work and overhead occurs
    - Partitioning can occur
- Studies have shown that a few stable nodes, then many “ephemeral” visits
- Solution:
  - divide population into 2 groups
    - Stable nodes (Super peers) with responsibility
    - Others, who can come and go with little structural impact
  - On structured P2P: keep republishing keys on a periodic basis

Churn (2)


SP 1
P 9
P 1
P 2
P 8
SP 3
SP 2
P 7
P 6
P 3
SP 4
P 5
P 4


P 6
P 5
SP 1
P 1
SP 2
P 2
SP 4
P 4
SP 3
P 3

Outline

- Definition
- Unstructured P2P systems
- Structured P2P systems
- Some applications
- Churn
- (Some) Security issues
- Conclusions

(Some) Security Issues

- In general, P2P system assume full cooperation between nodes
- Plenty of opportunity for an attacker to abuse the system:
  - denial-of-service attack
    - Refuse to route, denies existence of object, mis-routes
      - Shadow network
  - Pollution attack
    - Replies to everything with garbage
  - Man-in-the-middle attack
    - By definition it is easy
    - Can modify requests
  - False routing update info
    - To attract/repulse query traffic
  - Churn attack
  - Cybil attack
    - Physical node assumes very many virtual identities
    - Amplifies other attacks

Conclusions

- P2P systems are based on very few fundamental principles
  - Object location, indexing, storage, retrieval
  - Many applications can be built on these
  - Structured P2P focus on the objects – but may look very unstructured at IP level
- Often, nodes in a P2P systems are considered “homogeneous”
  - Equal functionality
  - Equal performance
  - Practice has shown that this is rarely the case
- This poses many performance and security issues
  - These are not insurmountable, but very often there is no provision to support solutions
- Trust propagation models can help in certain cases
  - But what if nodes can change identities

Thank you for your attention!

DOGBERT'S COMMUNICATION SEMINAR
THERE'S REALLY NO POINT IN LISTENING TO OTHER PEOPLE.
www.dilbert.com scottadams@aol.com
THEY'RE EITHER GOING TO BE AGREEING WITH YOU OR SAYING STUPID STUFF.
9-30-06 © 2006 Scott Adams, Inc./Dist. by UFS, Inc.
THAT SHOULD CUT DOWN ON THE QUESTIONS.

© Scott Adams, Inc./Dist. by UFS, Inc.