Peer-to-Peer Networks

06 Tapestry

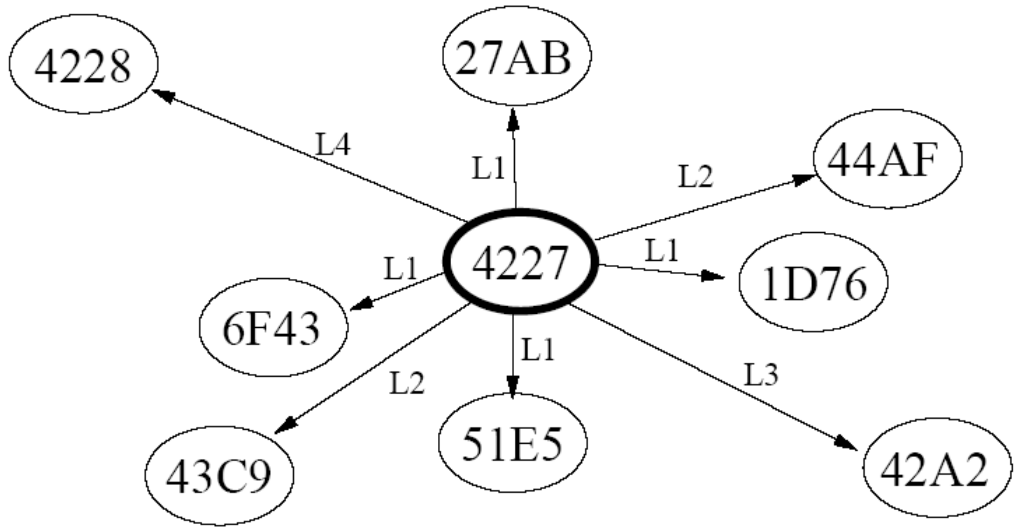
Christian Schindelhauer
Technical Faculty
Computer-Networks and Telematics
University of Freiburg

Tapestry

- Objects and Peers are identified by
  - Objekt-IDs (Globally Unique Identifiers GUIDs) and
  - Peer-IDs
- IDs
  - are computed by hash functions
    - like CAN or Chord
  - are strings on basis B
    - B=16 (hexadecimal system)

Neighborhood of a Peer (1)

- Every peer A maintains for each prefix x of the Peer-ID
  - if a link to another peer sharing this Prefix x
  - i.e. peer with ID B=xy has a neighbor A, if xy´=A for some y, y´
- Links sorted according levels
  - the level denotes the length of the common prefix
  - Level L = |x|+1



Neighborhood Set (2)

- For each prefix x and all letters j of the peer with ID A
  - establish a link to a node with prefix x_j within the neighboorhood set $N_{x,j}^{A}$
- Peer with Node-ID A has b |A| neighborhood sets
- The neighborhood set of contains all nodes with prefix s_j
  - Nodes of this set are denoted by (x,j)

Example of Neighborhood Sets

Neighborhood set of node 4221

	Level 4	Level 3	Level 2	Level 1
j=0	4220	420?	40??	0???
j=1	4221	421?	41??	1???
.	4222	422?	42??	2???
.	4223	423?	43??	3???
.	4224	424?	44??	4???
.	4225	425?	45??	5???
.	4226	426?	46??	6???
j=7	4227	427?	47??	7???

Links

- For each neighborhood set at most k Links are maintained

$$k \geq 1 : \left| N_{x,j}^{A} \right| \leq k$$

- Note:
  - some neighborhood sets are empty

Properties of Neighborhood Sets

- Consistency
  - If $N^{A}_{x,j} = \emptyset$ für any A
    - then there are no (x,j) peers in the network
    - this is called a hole in the routing table of level |x|+1 with letter j
- Network is always connected
  - Routing can be done by following the letters of the ID $b_1 b_2 \ldots b_n$

$$N^{A}_{\phi, b_1}$$ 1st hop to node A_1

$$N^{A_1}_{b_1, b_2}$$ 2nd hop to node A_2

$$N^{A_2}_{b_1 o b_2, b_3}$$ 3rd hop to node A_3

…

Locality

- Metric
  - e.g. given by the latency between nodes
- Primary node of a neighborhood set $N_{x,j}^{A}$
  - The closest node (according to the metric) in the neighborhood set of A is called the primary node
- Secondary node
  - the second closest node in the neighborhood set
- Routing table
  - has primary and secondary node of the neighborhood table

Root Node

- Object with ID Y should stored by a so-called Root Node with this ID
- If this ID does not exist then a deterministic choice computes the next best choice sharing the greatest common prefix

Surrogate Routing

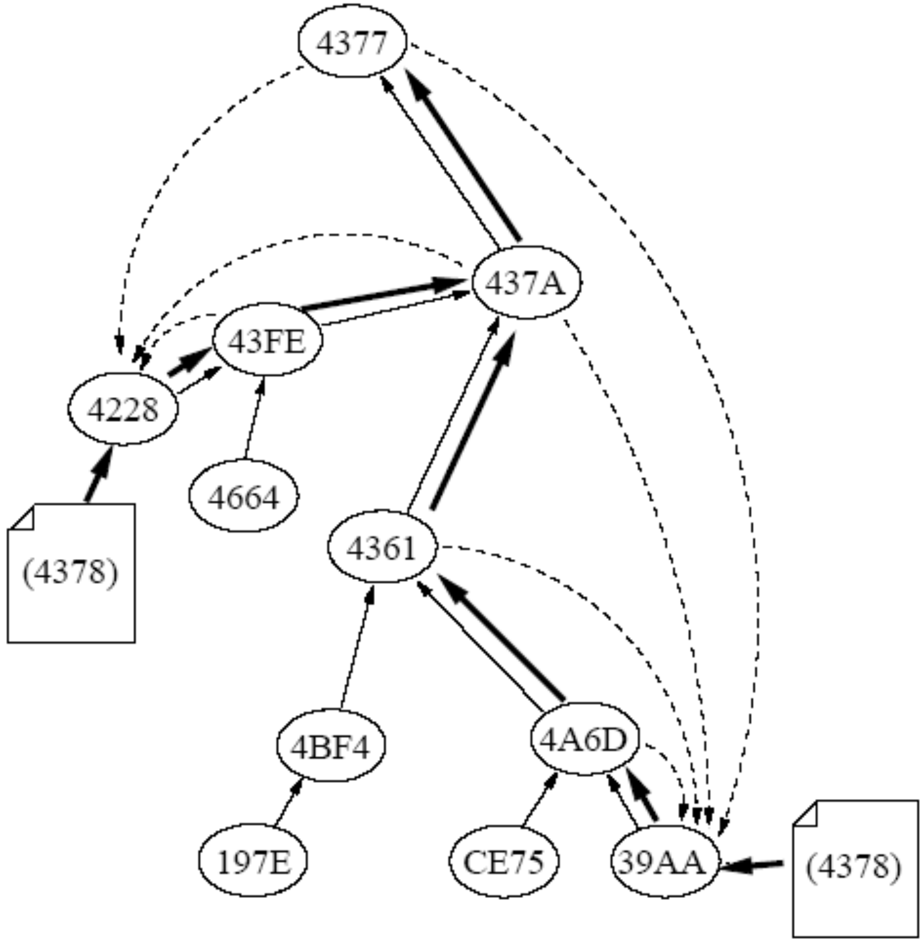
- Surrogate Routing
  - compute a surrogate (replacement root node)
  - If (x,j) is a hole, then choose (x,j+1),(x,j+2),… until a node is found
  - Continue search in the next higher level

Example: Surrogate Routing

- Lookup of 4666 by peer 2716

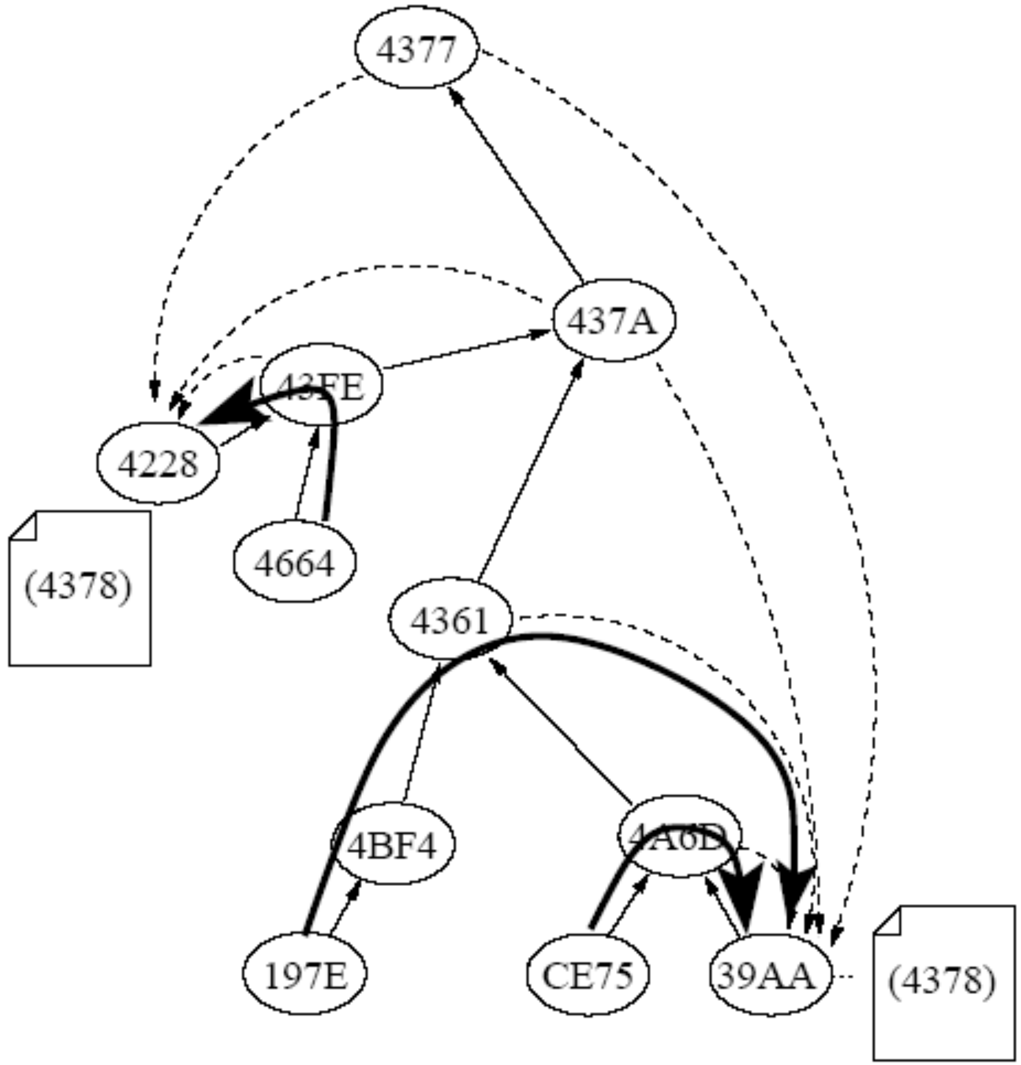
Publishing Objects

- Peers offering an object (storage servers)
  - send message to the root node
- All nodes along the search path store object pointers to the storage server



Lookup

- Choose the root node of Y
- Send a message to this node
  - using primary nodes
- Abort search if an object link has been found
  - then send message to the storage server



Fault Tolerance

- Copies of object IDs
  - use different hash functions for multiple root nodes for objects
  - failed searches can be repeated with different root nodes
- Soft State Pointer
  - links of objects are erased after a designated time
  - storage servers have to republish
    - prevents dead links
    - new peers receive fresh information

Surrogate Routing

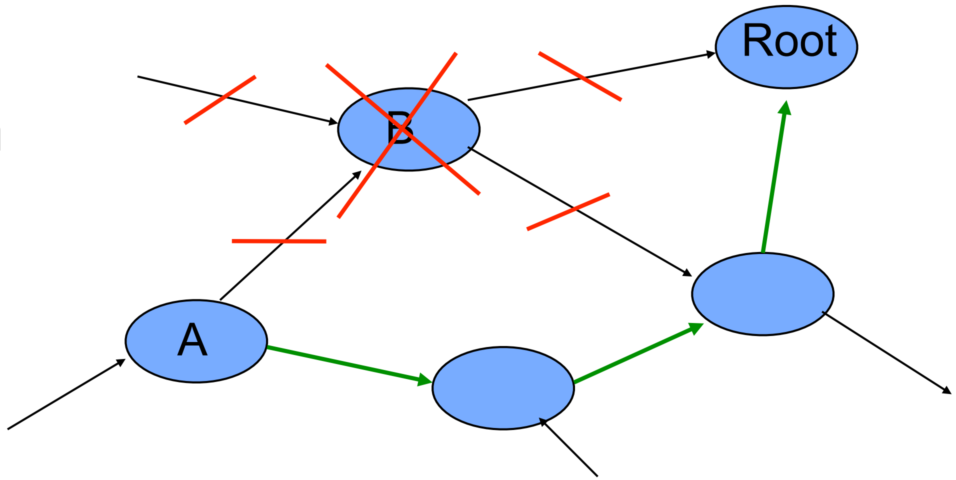
- Theorem
  - Routing in Tapestry needs O(log n) hops with high probability

Adding Peers

- Perform lookup in the network for the own ID
  - every message is acknowledged
  - send message to all neighbors with fitting prefix,
    - Acknowledged Multicast Algorithm
- Copy neighborhood tables of surrogate peer
- Contact peers with holes in the routing tables
  - so they can add the entry
  - for this perform multicast algorithm for finding such peers

Leaving of Peers

- Peer A notices that peer B has left
- Erase B from routing table
  - Problem holes in the network can occur
- Solution: Acknowledged Multicast Algorithm
- Republish all object with next hop to root peer B



Pastry versus Tapestry

- Both use the same routing principle
  - Plaxton, Rajamaran und Richa
  - Generalization of routing on the hyper-cube
- Tapestry
  - is not completely self-organizing
  - takes care of the consistency of routing table
  - is analytically understood and has provable performance
- Pastry
  - Heuristic methods to take care of leaving peers
  - More practical (less messages)
  - Leaf-sets provide also robustness

Peer-to-Peer Networks
06 Tapestry

Christian Schindelhauer
Technical Faculty
Computer-Networks and Telematics
University of Freiburg