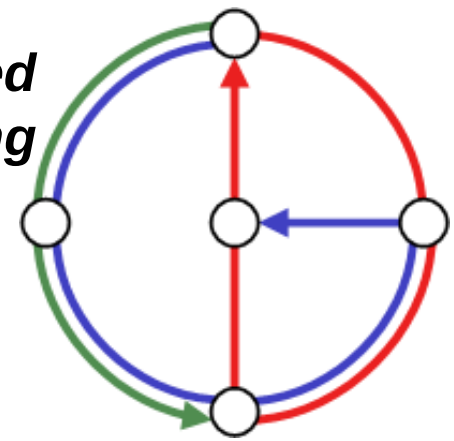
Aggregating Information in Peer-to-Peer Systems for Improved Join and Leave

Distributed Computing Group



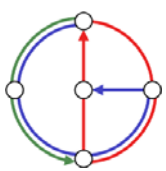
Keno Albrecht
Ruedi Arnold
Michael Gähwiler
Roger Wattenhofer

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

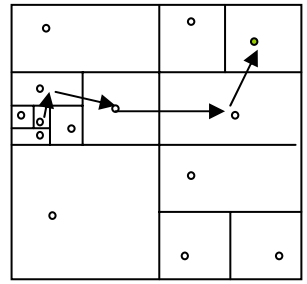
P2P2004

Overview

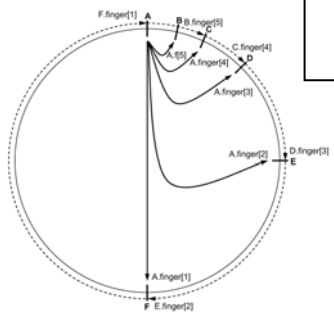
- Introduction
  - P2P Topologies: “Tree” Structure
  - Join & Leave in P2P
- Distributed Approximation System Information Service (DASIS)
- Join Algorithms using DASIS
- Simulation Results
- Conclusion & Outlook



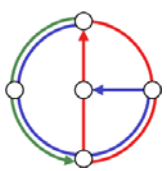
P2P Topologies



- Different P2P Topologies
  - Ring, d-dimensional address space…

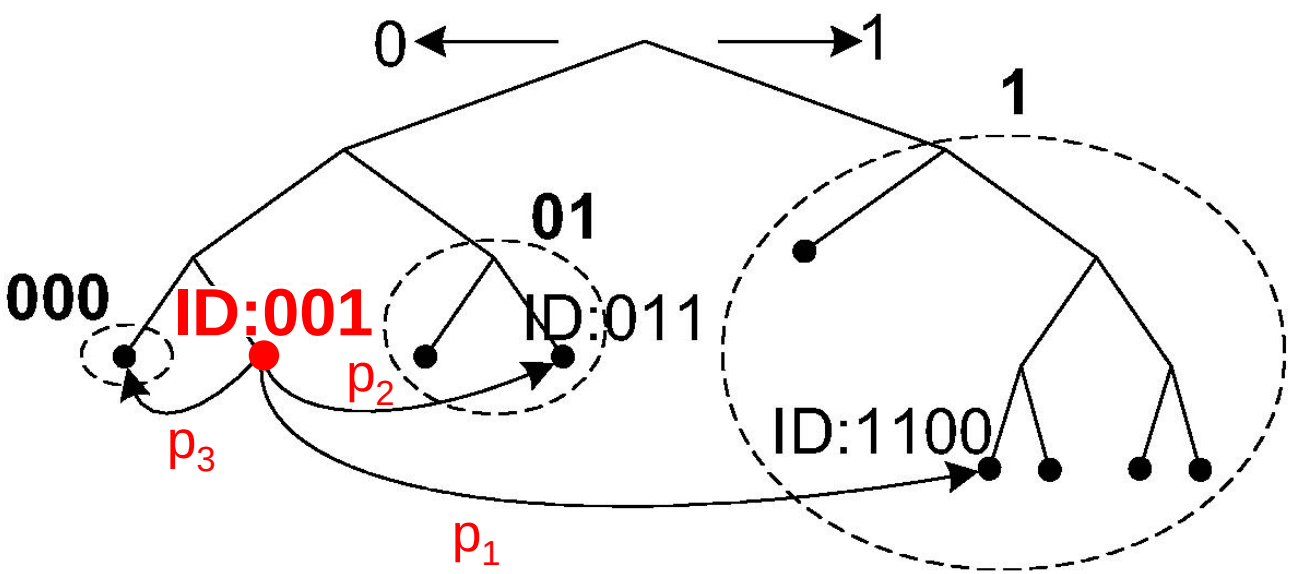


- **Tree Topology**
  - Unique ID (bit string) per peer
  - ID specifies “domain space”
  - Peer is responsible for storing all keys with IDs within its domain space (longest prefix match)
  - Example: **Kademlia** (P. Maymounkov and D. Mazieres. **Kademlia: A Peer-to-peer Information System Based on the XOR Metric.** In Proceedings of IPTPS, Cambridge, MA, USA, March 2002.)

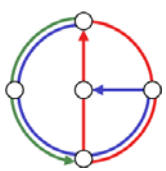


Tree P2P Topology

- Tree-Example:
  - Neighbors p_i
    - Neighbor p_i has the **same (i-1)** first bits and bit **i inverted**
    - Allow efficient **logarithmic search** (lookup) operation
  - Consider peer with ID 001

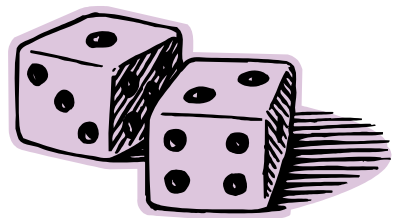


Peer 001 Neighbors	
i	**neighbor**
1	1100
2	011
3	000



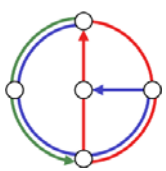
Joins and Leaves in P2P Systems

- Lingering Problem: **Assignment of ID** to peers
- P2P systems are decentralized and dynamic: **random** assignment is used
  - Used as reference in simulations



- This does **not well balance** the peers
  - Logarithmic imbalance factor (balls-into-bins analysis)
  - A highly loaded peer stores a factor $\Theta(\log n)$ more keys than a peer with average load, whp.
- Different random remedies have been proposed

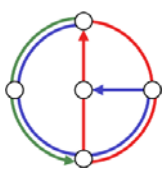
- Our proposal: a **non-randomized join-algorithm**
  - based on a distributed approximative information service for P2P systems



DASIS

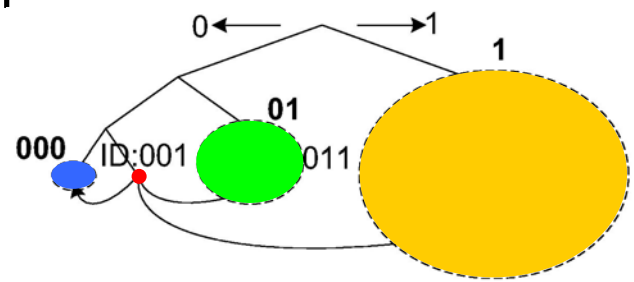
Distributed Approximation System Information Service

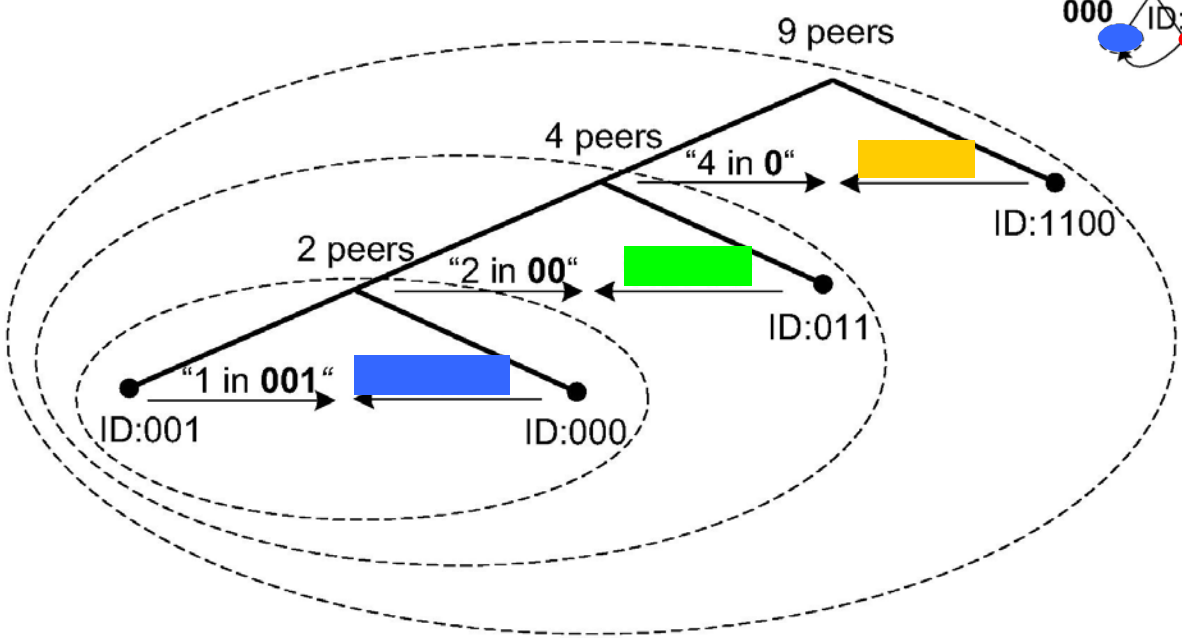
- Abstract decentralized service
- Provides approximate information about the P2P system
- Built on top of the regular P2P structure
- Accuracy depends on the message propagation mechanism
- Can deliver a wide range of information, e.g.:
  - number of peers in the system (illustrative example)
  - **minimal depth of peers in the system**
  - average up-time
  - total amount of bytes stored in the system
  - …



DASIS Example: Number of Peers

- Main idea: peer p is considered an "**expert**" on all the **sub domains** of the prefixes of its bit string ID $b_1b_2…b_n$
- The expert knowledge is constructed **inductively** through information exchange with the neighbor peers
- Let's consider our example again…

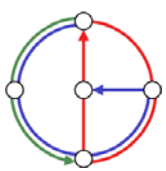




DASIS - accuracy

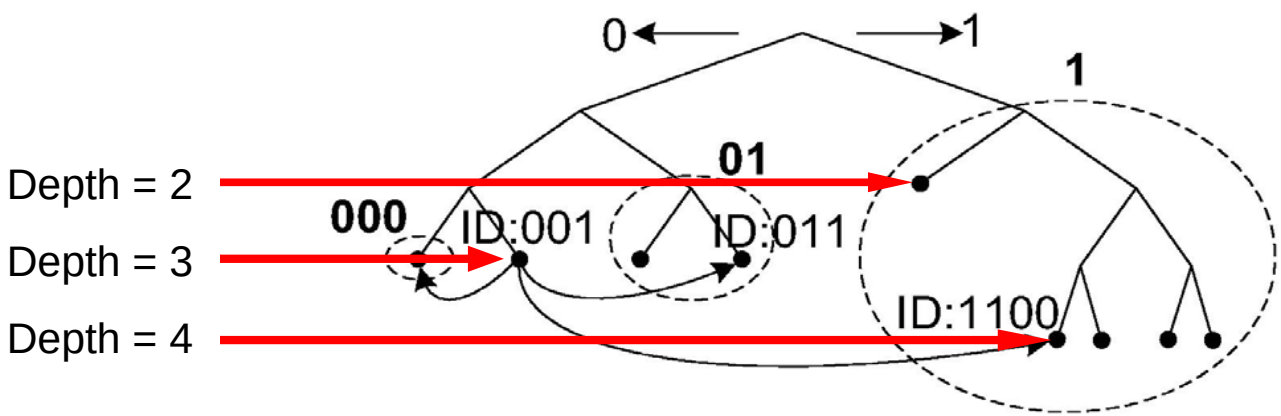
- For the **implementation** of DASIS, several update strategies were employed
  - **Periodical:** regular updates to neighbors
  - **Adaptive**: only if changes occurred
  - **Piggyback**: along with regular (e.g. lookup) messages

- Favorite: **Piggybacking** - no additional (message) cost!
  - Quality reduces gracefully



Join Algorithms using DASIS

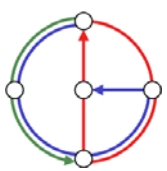
- Idea: **insert** peer where they are “most needed”
  - How is this need specified?
- We want a **well-balanced** topology!
  - Assumption: large and uniform distributed key population
  - All Peers should be at more or less the same depth
    - **Depth** of a peer = length of its bit string



Depth Join

- Uses DASIS minimal depth service

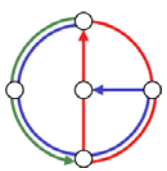
- **Algorithm**
  - New peer is **routed** through the P2P system to sub domains with smallest depth
  - At every peer passed, **one bit** of the bit string of the “joiner” is **fixed** -> Termination guaranteed
  - If further routing is not possible -> **insert**
    - Inserting peer assigns joining peer its own bit string plus a 1 and appends a 0 to its own bit string
    - Splitting the local domain space in half



Simulation – the Criterion

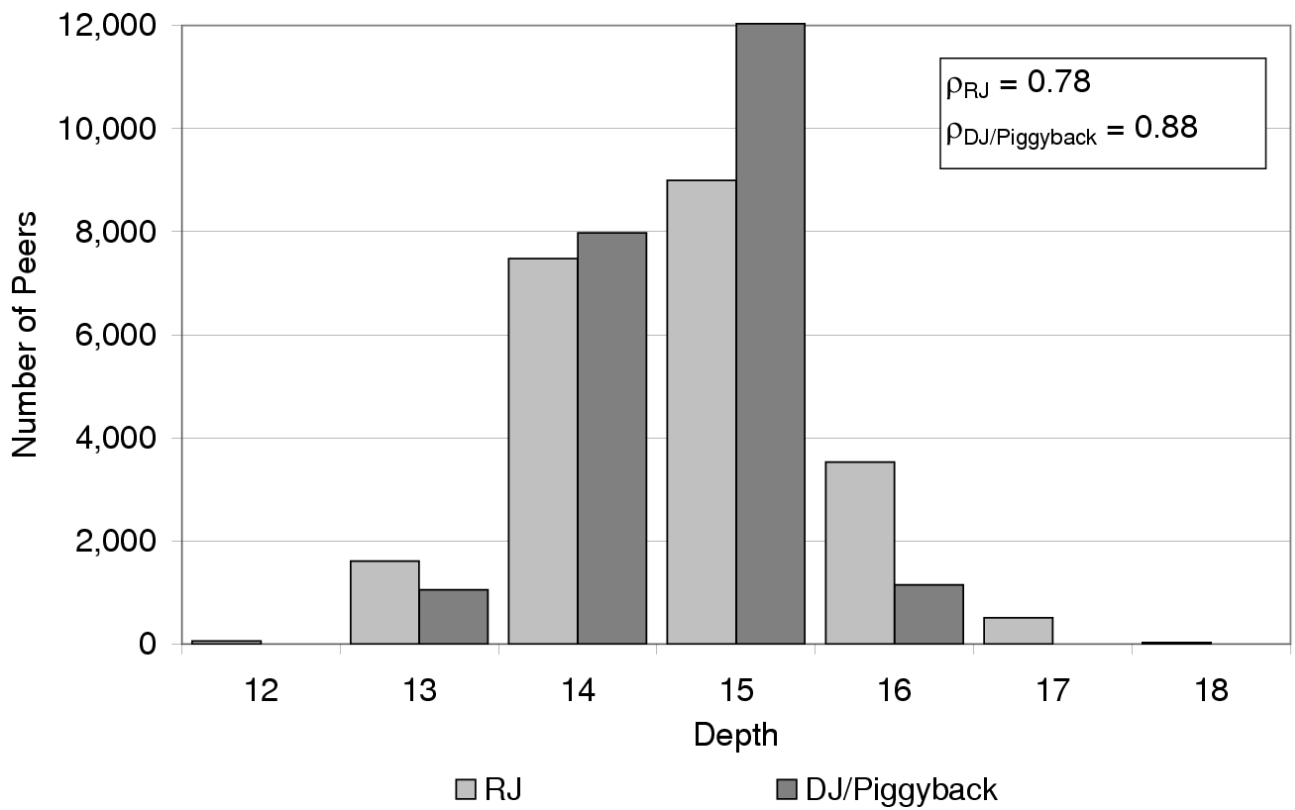
- Two evaluation **criteria**
  - minimal depth D
  - Balance measure B
    - More fine-grained
    - taking the number of peers with small depth into account
    - $B = \sum 2^{-2d_i} > 0$
      - Sum over all peers in the system
      - d_i = depth of peer i
    - Normalized with optimal balance B_{Opt}: $\rho_{\mathbf{Alg}} = B_{Opt} / B_{Alg} > 0$
      - $\rho_{\mathbf{Opt}} = 1$

- Random Join (RJ) for reference



Simulation – Inserting 22k Peers

- Depth Join (DJ) performs better than Random Join (RJ)
  - minimal depth service only piggybacking regular messages

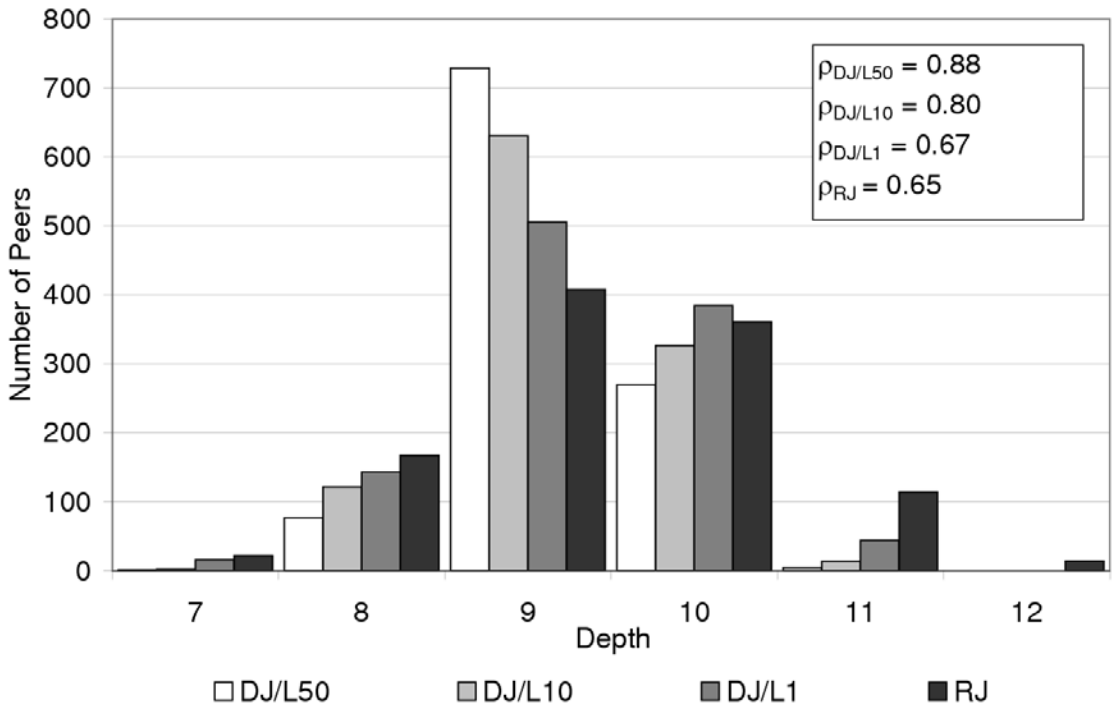


Simulation – Steady State

- Peers joining, leaving and performing lookup operations
  - **Load parameter L**: average number of lookups before leaving
  - Lookup messages for piggybacking
  - Higher load -> more accurate depth information -> better balanced system

- steady state is quickly reached!
  - Independent of start distribution

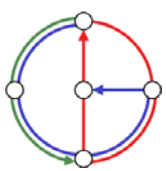
- Better than RJ



Conclusion

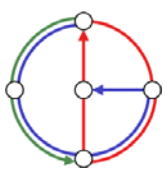
- We introduced DASIS the Distributed Approximation System Information Service
  - Powerful tool, many applications
  - Sample application: Minimal depth information for improved Join and Leave

- Join Algorithm based on depth information leads to better balanced P2P systems
  - Better than random assignment of IDs (Random Join)
  - Especially in cases of imbalances (e.g. due to a high number of leaving peers, or malicious attackers)



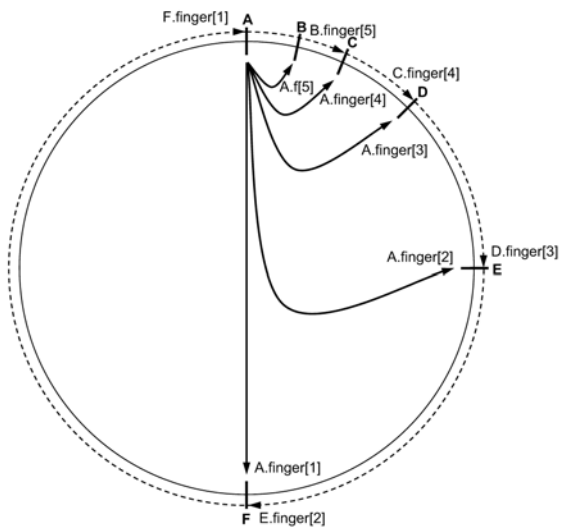
Another Conclusion

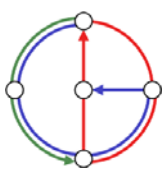
- Similar ideas contemporaneously developed for “Willow”
  - follow-up work of Astrolabe
  - Support for publish/subscribe
  - Uses SQL-like queries for information aggregation
  - Not employed for an improved join algorithm
  - Shows variety of applications of aggregation mechanism.
  - (R. van Renesse and A. Bozdog. Willow: DHT, Aggregation, and Publish/Subscribe in One Protocol. In Proceedings of IPTPS 2004.)



Outlook 1

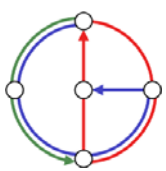
- DASIS was explained on a “Tree Style“ P2P topology
- Could also be integrated into existing P2P systems
  - Skip List based topologies such as Skip Graph, SkipNet
  - Ring topologies like Chord or Pastry
    - Sketch of DASIS on Chord in the paper
    - “How many peers in interval” instead of “How many peers with prefix x”
    - Finger intervals may overlap
      - Adjustment approximately possible
      - No problem for min or max functions





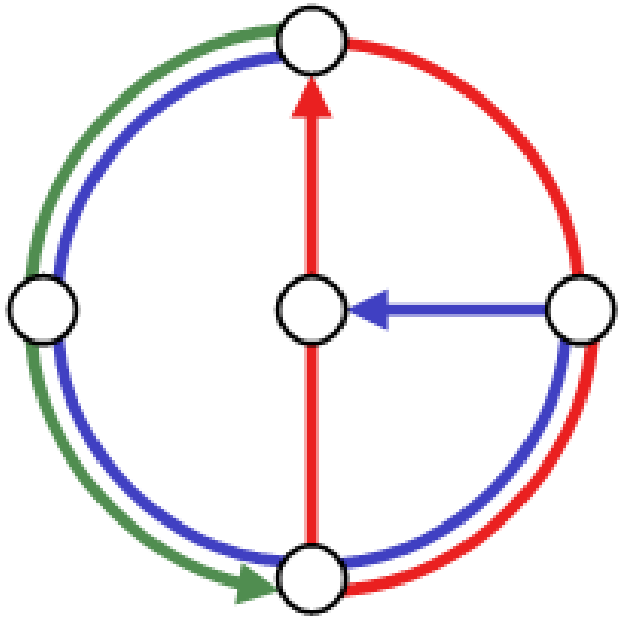
Outlook 2

- Theoretical analysis in progress…
  - To be published soon by Kuhn, Schmid and Wattenhofer
  - Surprising results
    - Synchronous model
    - system with diameter D
    - ⇨ “Every node knows exact state of system D time ago”
- Still to be done…
  - Implement DASIS and Join Algorithm for real-world measurement



Questions?
Comments?

***D**istributed*
***C**omputing*
***G**roup*



Thank you. ☺