PeerMart: The Technology for a Distributed Auction-based Market for Peer-to-Peer Services

David Hausheer

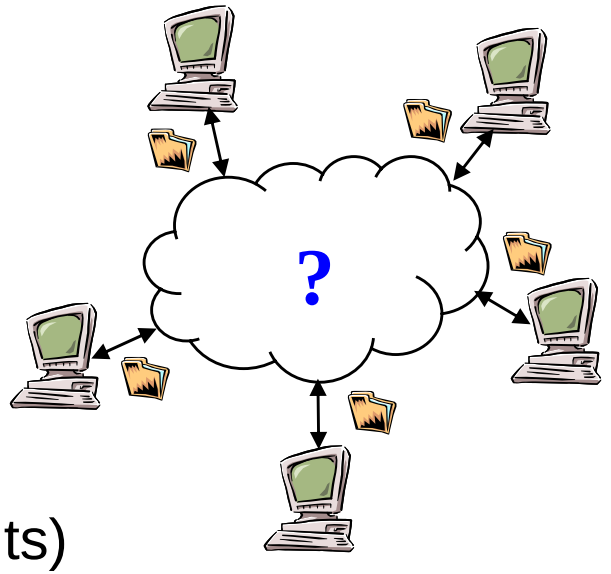
Computer Engineering and Networks Laboratory TIK
Swiss Federal Institute of Technology ETH Zurich
E-Mail: hausheer@tik.ee.ethz.ch

Institut für Technische Informatik und Kommunikationsnetze

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Motivation

- Key question
  - How to build an economically efficient and technically feasible market place for users to trade goods or services over the Internet?



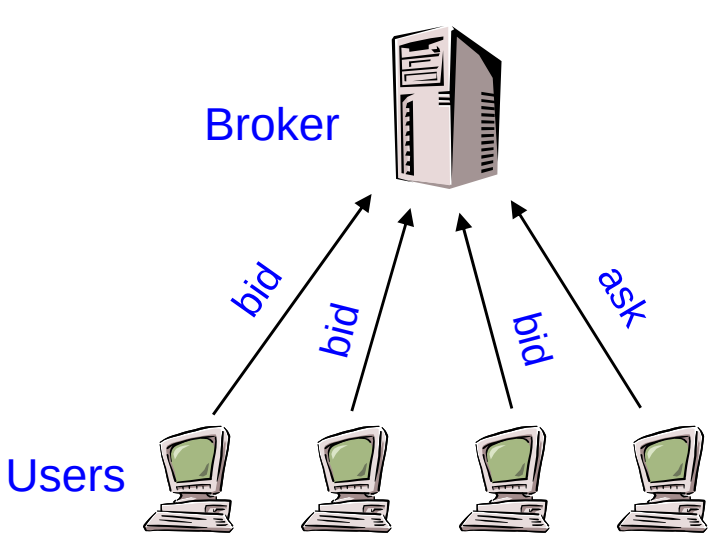
- Economic approach
  - Use of auctions
    - Well established concept
    - Economically efficient (=> maximal benefits)

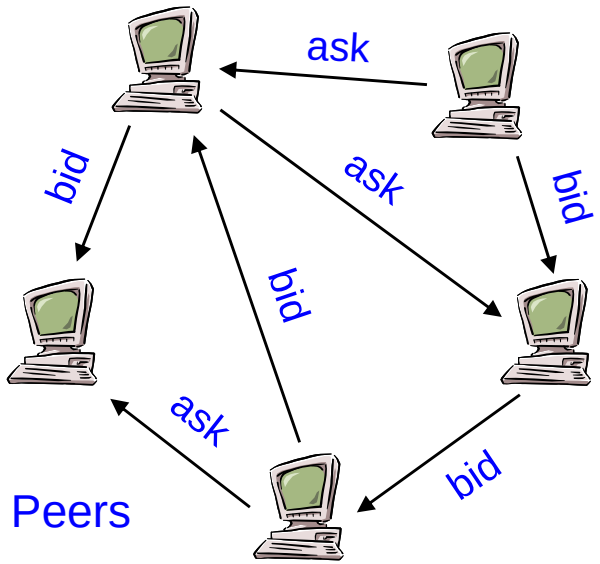
- Key technical requirements
  - Scalability (=> scale to any number of users)
  - Reliability (=> correctness, continuous availability, security)
  - Technical efficiency (=> efficient use of resources, i.e. CPU, storage, network links)

Centralized versus Decentralized Auctions

- Centralized (C/S) auctions
  - Dedicated (centralized) broker (e.g. eBay)
  - \+ High reliability
  - \+ High technical efficiency
  - \- Single Point of Failure
  - \- Vulnerable against attacks
  - \- Not scalable

- Decentralized (P2P) auctions
  - Peers act as brokers, providers, and consumers at the same time
  - \+ No Single Point of Failure
  - \+ Scalable
  - \- Vulnerable against selfish or malicious peers
  - \- Technical feasibility?





Goals and Outline

- Goals
  - Maintain the economic efficiency of auctions
  - Exploit the scalability and resilience of P2P networks
  - Achieve a high reliability even in the presence of malicious peers
  - Provide a technically efficient solution

=> Focus on technical feasibility of implementing a P2P auction

- Outline
  - PeerMart‘s approach and technical design
  - Analytical and experimental results

PeerMart Approach

- Key Idea
  - Distributed double auctions
  - Provider and consumer peers offer prices for services
  - Offers are optimally matched by peers acting as brokers for a service
- Basic Algorithm
  - Providers publish services they wish to provide
    - Broker peers reply bid price (current highest pay price)
  - Consumers request services they wish to use
    - Broker peers reply ask price (current lowest sell price)
  - Brokers run the following matching strategy at regular intervals:
    - Upon every price offer, if offer is higher (lower) than current bid price (ask price) => no match, store offer in a table
    - Otherwise, forward offer to the peer with the best counteroffer (highest bid / lowest ask)

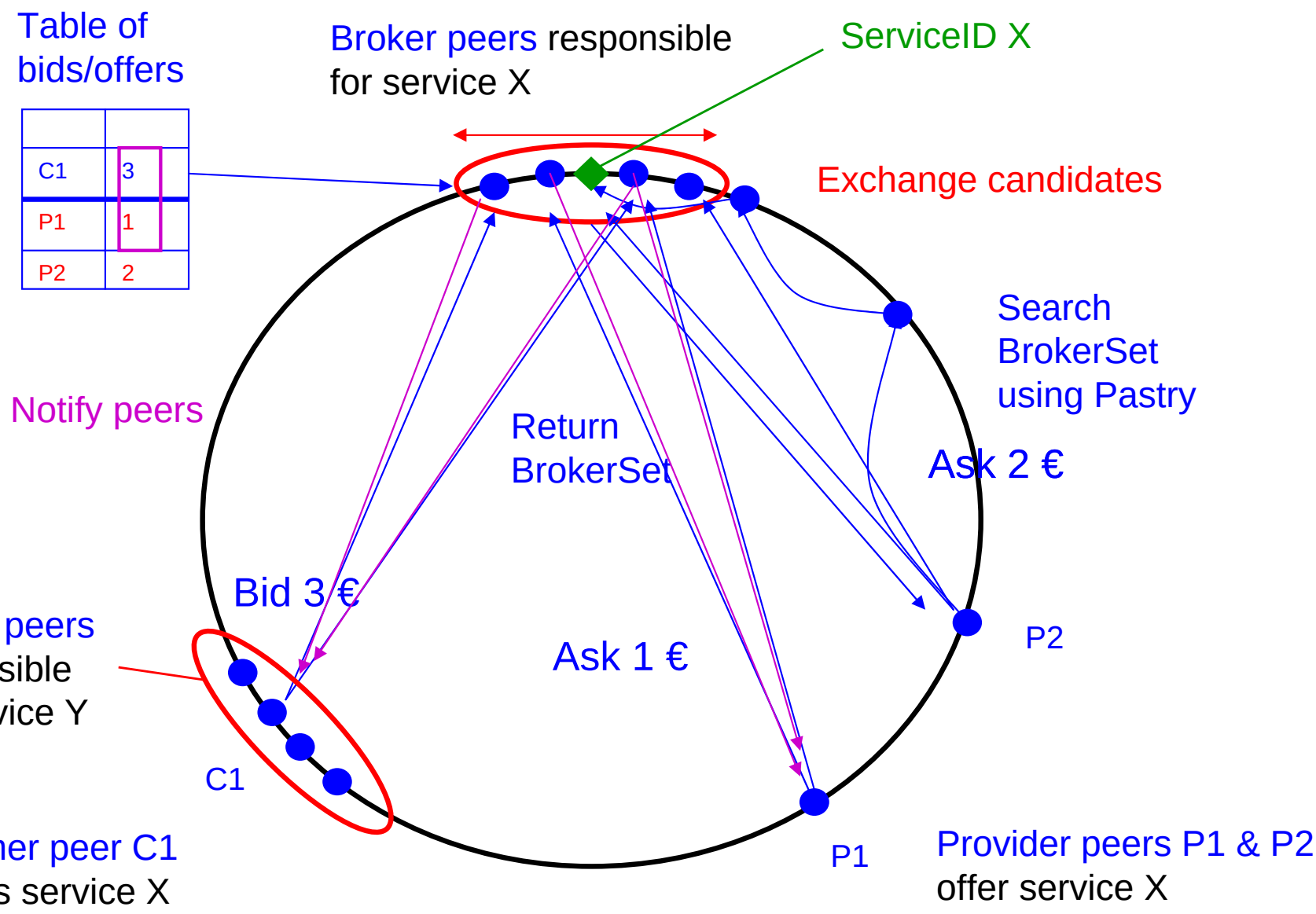
Technical Design

- PeerMart uses a structured P2P overlay network to distribute broker load
  - Pastry is used as underlying infrastructure
    - Provides unique 128-bit nodeIds
    - Stores *n* closest nodeIds (leaf-set) in routing table
- Each peer has a public / private key pair to sign messages
  - Assumption: Public keys certified offline by trusted entity and bound to nodeIds
- Services are mapped onto a redundant set of *n* broker peers (broker-set)
  - Assumption: Services have unique IDs (e.g. hash value of a file)

Technical Design (cont'd)

- Consumers/providers randomly select *f* brokers out of the broker set
  => Load is uniformly distributed on all broker peers
- Each broker peer keeps table of *m* highest bids / lowest asks
  - Lower bids and higher asks are rejected
- Matching is performed in a decentralized way
  - Candidates for a match are forwarded to other brokers
    – Candidates: local matches and next best offer
  - Final matches are determined using majority decisions
  - Winners are notified by the *f* brokers that received the offer

Double Auction in PeerMart



Analytical Results

- Basic effort for maintaining Pastry is $O(\log_b N)$
  - N: Number of peers in the network
- Finding responsible broker-set is $O(\log_b N)$
- All subsequent communication is direct
  - Sending offers is O(f)
  - Exchange/forward of matching candidates is O(f*n)
  - Notify peers about final matches is O(f)

  } Total costs
  - f: Number of parallel brokers used
  - n: Number of peers in a broker set
- Avg. number of offers stored per peer: s*n*m
  - s: Avg. number of service involvements per peer (naturally limited)
  - m: Table size for storing offers (limited)

Experiment Setup

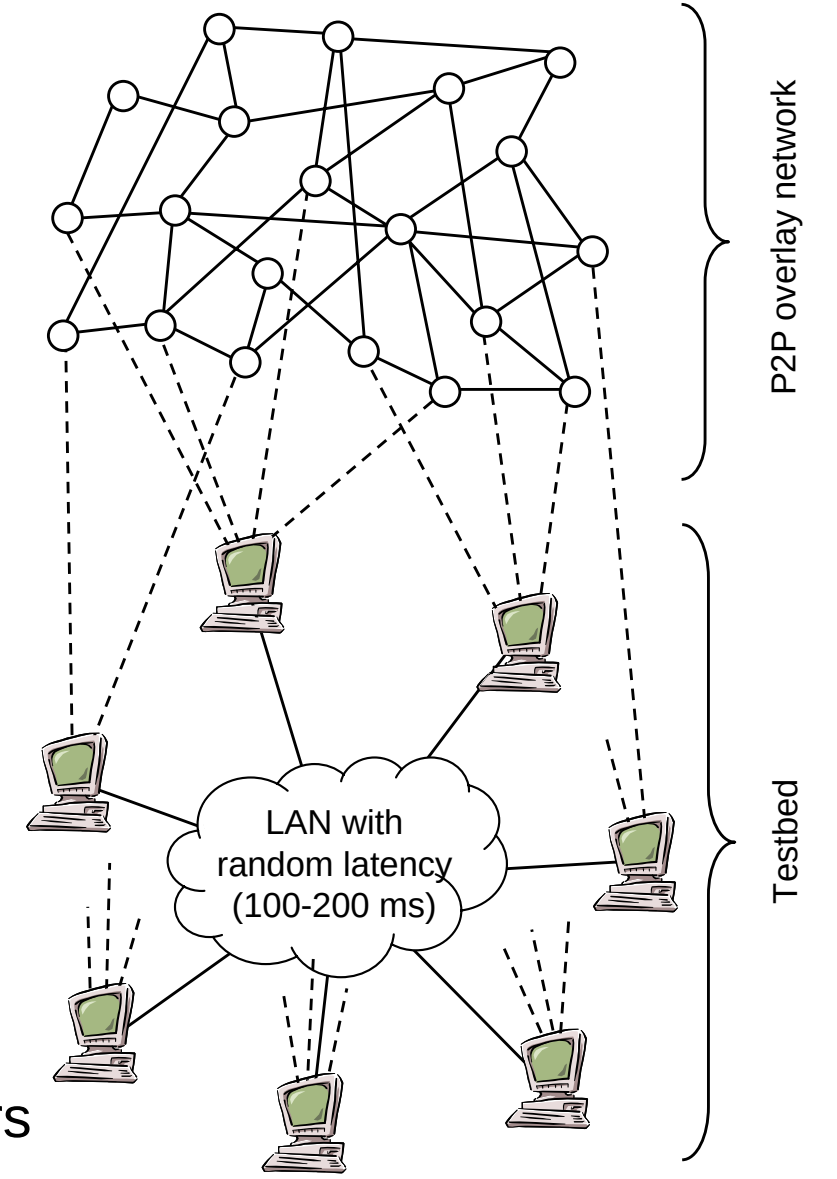
- Prototype implemented in Java on top of FreePastry
- Assumptions for experiments:
  - Each peer assigned to *s* = 2 services
  - Peers follow the ZIP bidding strategy
    - Consumer's price offer:

$$p_{bid} = min(p_{ask} + \alpha(p_{max} - p_{ask}), p_{max})$$

    - Provider's price offer:

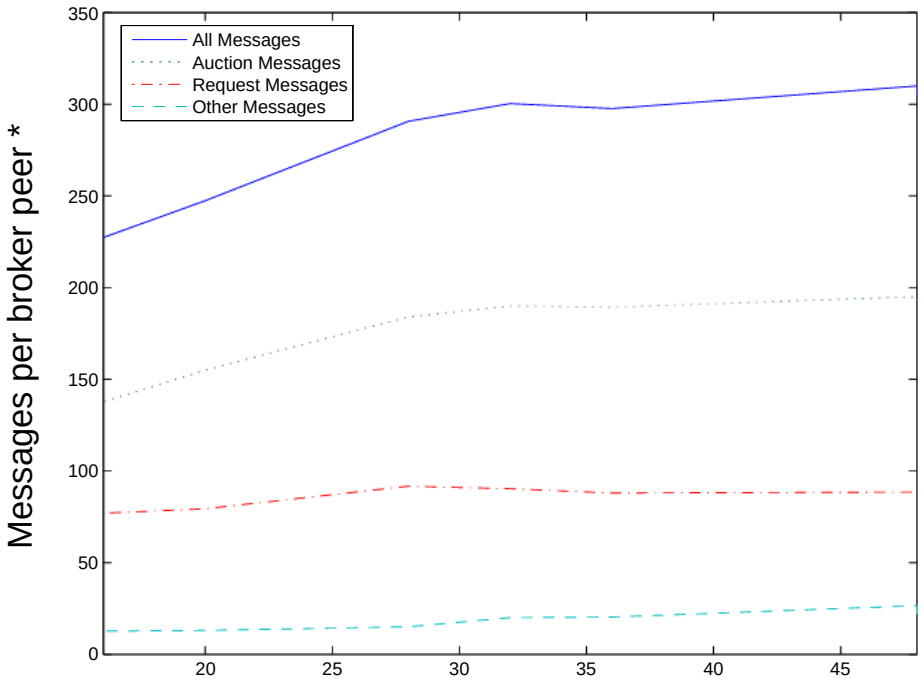
$$p_{ask} = max(p_{bid} - \alpha(p_{bid} - p_{min}), p_{min})$$

    - p_{max}, p_{min} are reservation prices => normally distributed such that 50% of offers match, α set to 0.1
  - Malicious peers
    - Uniformly distributed
    - Do not forward offers nor notify peers

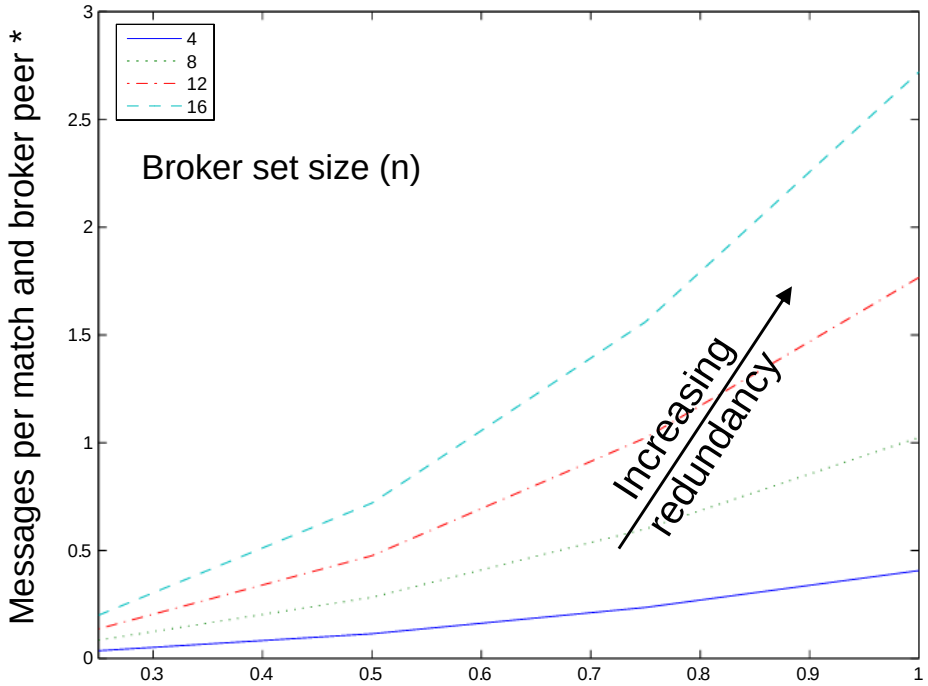


Experimental Results

Scalability



Efficiency (redundancy costs)

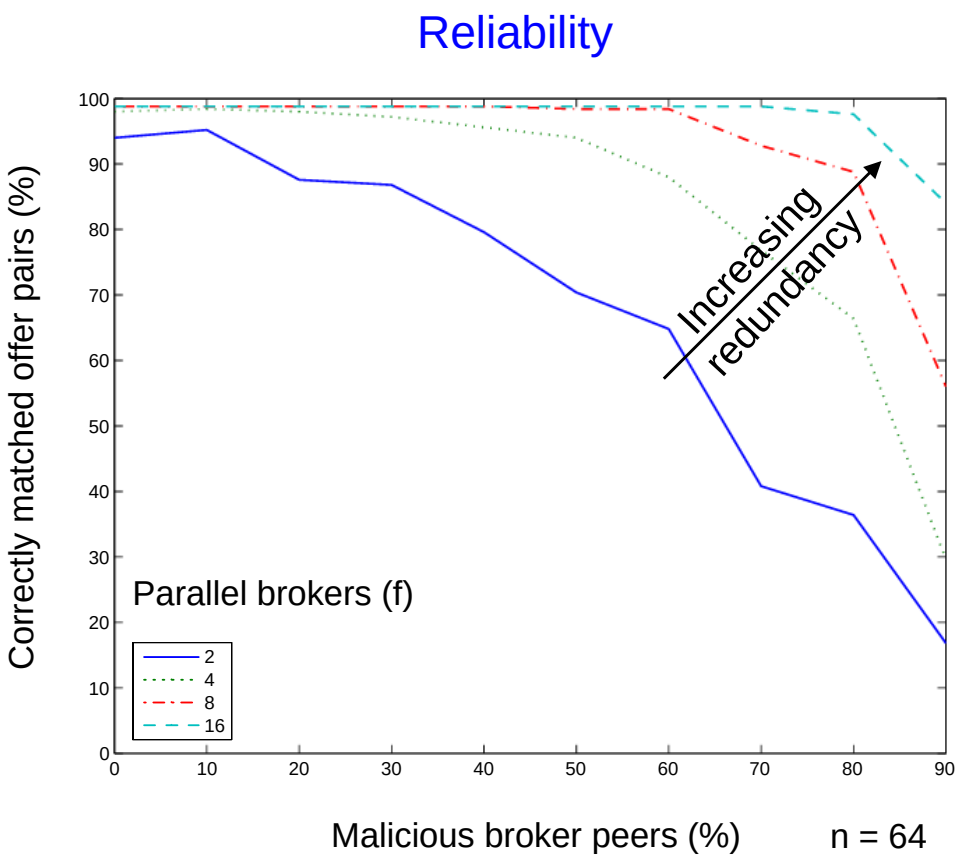


Measured security overhead:

- Overhead for SHA1 with DSA signature is ~15% of message size
- Signing / verifying messages takes 10 - 20 ms (~10% of the average RTT) on Pentium 4 CPU 2.4 GHz, 512 MB RAM, Java VM 1.4.2

* Message size is ~1 kB

Experimental Results (cont'd)



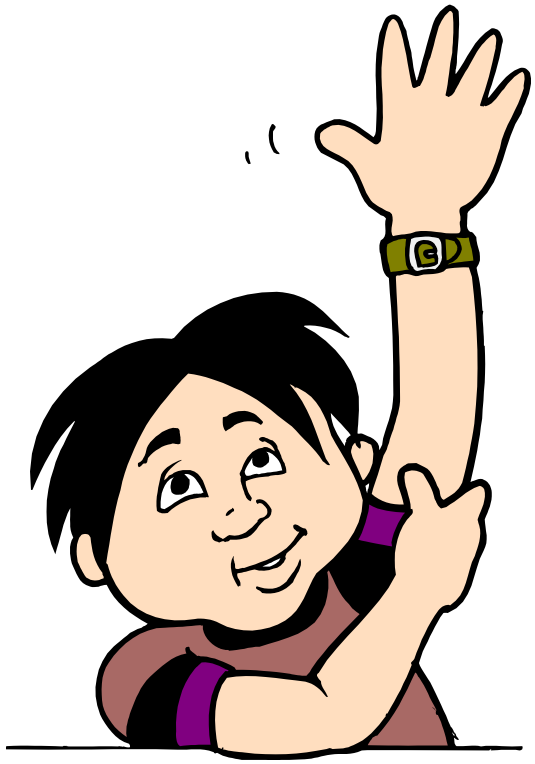
- PeerMart correctly matches offer pairs for < 50% malicious peers using, *e.g.*, 8 brokers in parallel (out of 64 brokers in total)

- If fraction of malicious peers is < 25%, even 4 parallel brokers provide good reliability

Conclusions and Future Work

- PeerMart is a fully distributed auction-based market
  - Can be used by peers to trade any services
- Technically designed on top of a P2P overlay network
  - Implemented prototype uses FreePastry as underlying infrastructure
- PeerMart is both technically and economically efficient
  - Scales well even for a large number of peers trading services
- High reliability even in the presence of malicious peers
  - Achieved through redundancy
- Future Work
  - Punish peers that do not stay to an offered price (=> use Reputation)
  - Study other forms of malicious attacks (e.g. DDoS)
  - Consider and prevent overlay splitting
  - Run prototype in a real-world environment (e.g. PlanetLab)

Thank you! – Any Questions?

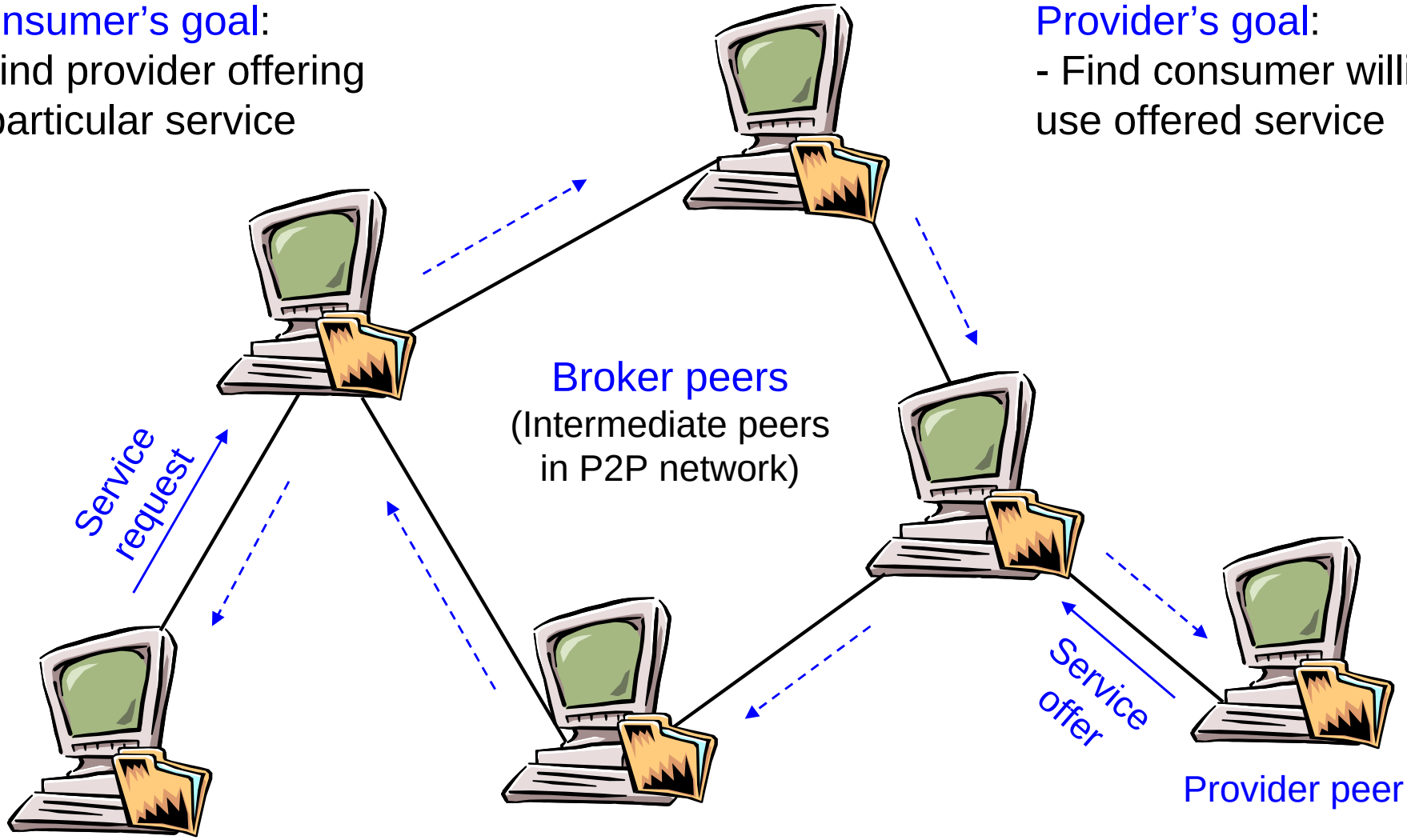


=> Further info: http://www.peermart.net/

P2P Market Model: Roles and Interactions

Consumer's goal:
- Find provider offering a particular service

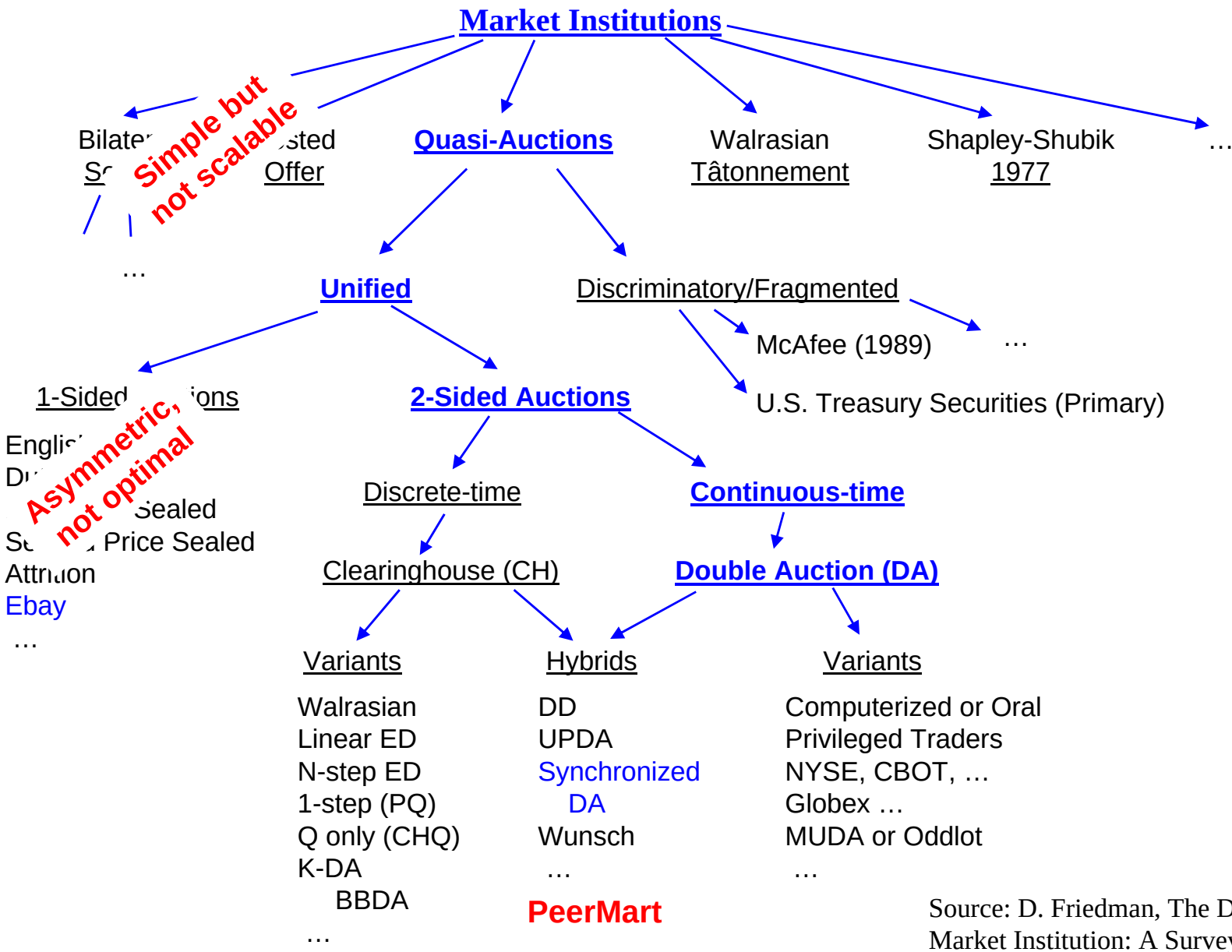
Provider's goal:
- Find consumer willing to use offered service



Decentralized Auctions: Main Problems

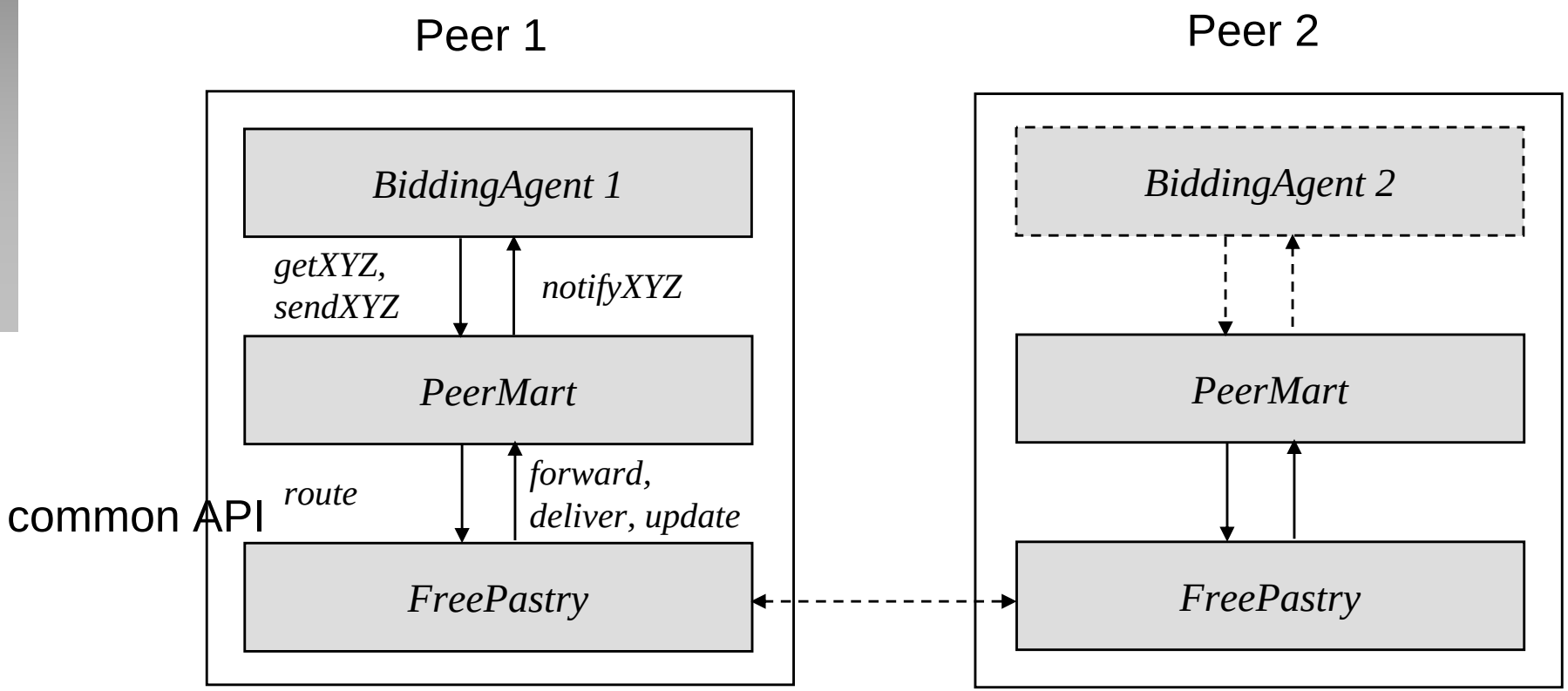
- Why should peers cooperate?
  - Forwarding messages (requests / bids / offers etc.)
  - Caching / Processing data on behalf of other peers
- Cooperation is costly
  - Peers have simply no incentives to cooperate
- Even worse: Peers might be competitors
  - Provider peers could lose opportunity to sell
  - Consumer peers could lose opportunity to buy
- Peers are autonomous
  - They may act in a selfish or malicious way
- Accounting for forwarding messages etc is not feasible
  - Difficult to measure, high technical effort

Auction-based Markets: Design Space



Source: D. Friedman, The Double Auction Market Institution: A Survey, 1993

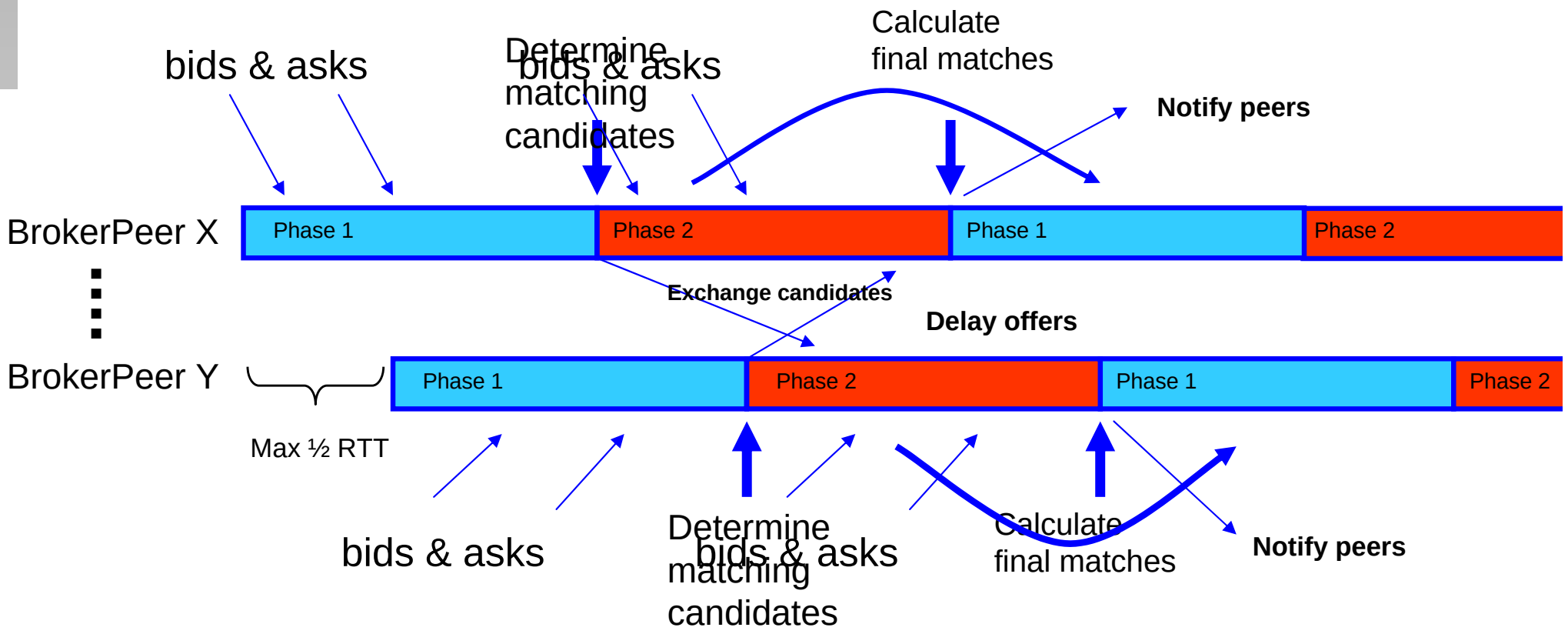
Implementation of PeerMart



Exchange Candidates

- Problem
  - Message delay between peers
  - Time is not synchronized

=> Introduce time slots greater than max Round Trip Time

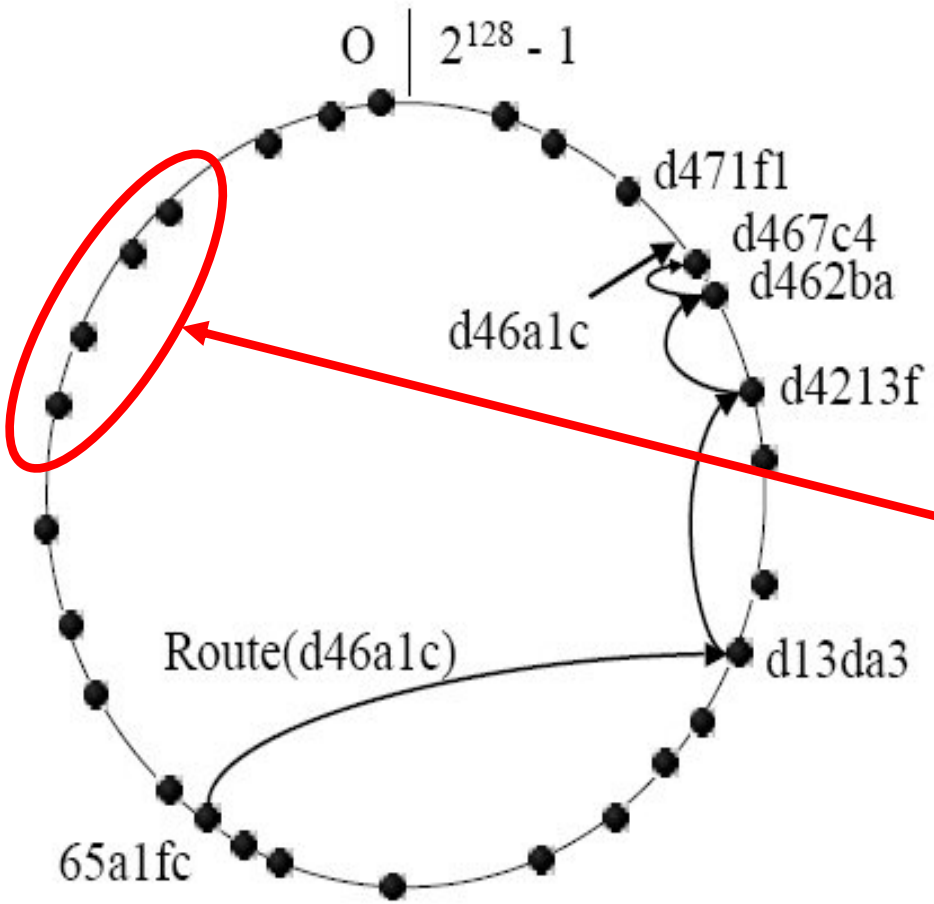


Related Work on P2P Auctions

- Z. Despotovic, J. Usunier, K. Aberer: Towards Peer-To-Peer Double Auctioning; In Proceedings of the 37th HICSS Conference, January 2004.
  - Offers/Bids are broadcasted in a Gnutella-like fashion
  - Any peer can answer with a counteroffer
  - Not scalable, not strategy-proof
- E. Ogston, S. Vassiliadis: A Peer-to-Peer Agent Auction; In Proceedings of AAMAS Conference, July 2002.
  - Agents are connected in a random network
  - Single agent is assigned as cluster center
  - Cluster size is limited => not scalable
  - No message delay is assumed => not realistic
- M. Fontoura, M. Ionesu, N. Minsky: Decentralized Peer-to-Peer Auctions; Journal of Electronic Commerce Research, January 2005.
  - Based on Law Governed Interaction (LGI) paradigm
  - Only the auction process itself is decentralized
  - Communication overhead due to routing messages via controllers
- Many more on double auctions (not P2P) ...

Underlying Infrastructure: Pastry

NodeId space



Prefix-based routing

Unique 128-bit nodeIds (calculated from a peers IP address or public key using secure hash function)

Open source implementation (FreePastry)

Routing Table

NodeId 10233102

Leaf set	SMALLER	LARGER	
10233033	10233021	10233120	10233122
10233001	10233000	10233230	10233232

Routing table

-0-2212102	1	-2-2301203	-3-1203203
0	1-1-301233	1-2-230203	1-3-021022
10-0-31203	10-1-32102	2	10-3-23302
102-0-0230	102-1-1302	102-2-2302	3
1023-0-322	1023-1-000	1023-2-121	3
10233-0-01	1	10233-2-32	
0		102331-2-0	
		2	

Neighborhood set

13021022	10200230	11301233	31301233
02212102	22301203	31203203	33213321