PeerMart: Pricing Mechanisms in Peer-to-Peer Systems

David Hausheer

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

Computer Engineering and Networks Laboratory,
Swiss Federal Institute of Technology, ETH Zürich

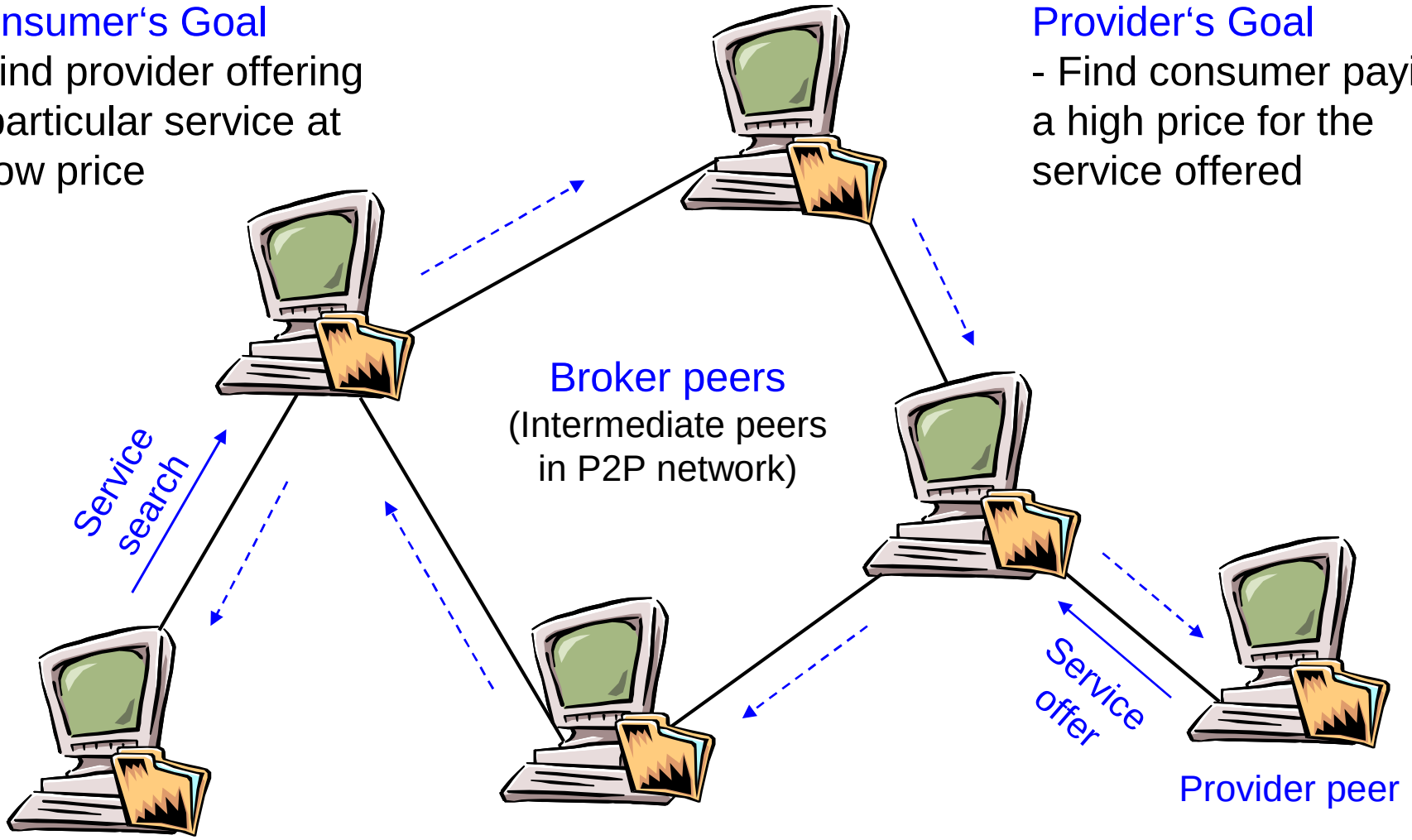
Introduction and Outline

- Goal
  - To develop a pricing mechanism for P2P services
    - Enabling the commercial use of P2P technology
    - Supporting applications going beyond best effort file sharing
    - Give peers an incentive to provide valuable services
- Requirements
  - Pricing mechanism needs to be
    - Economically efficient
    - Technically efficient and scalable
    - Reliable (robust against selfish/malicious behavior of peers)
- Outline
  - P2P Market
    - General model, main problems, design space for pricing
  - PeerMart‘s Approach
  - Evaluation

P2P Market: Basic Model and Roles

Consumer‘s Goal
- Find provider offering a particular service at a low price

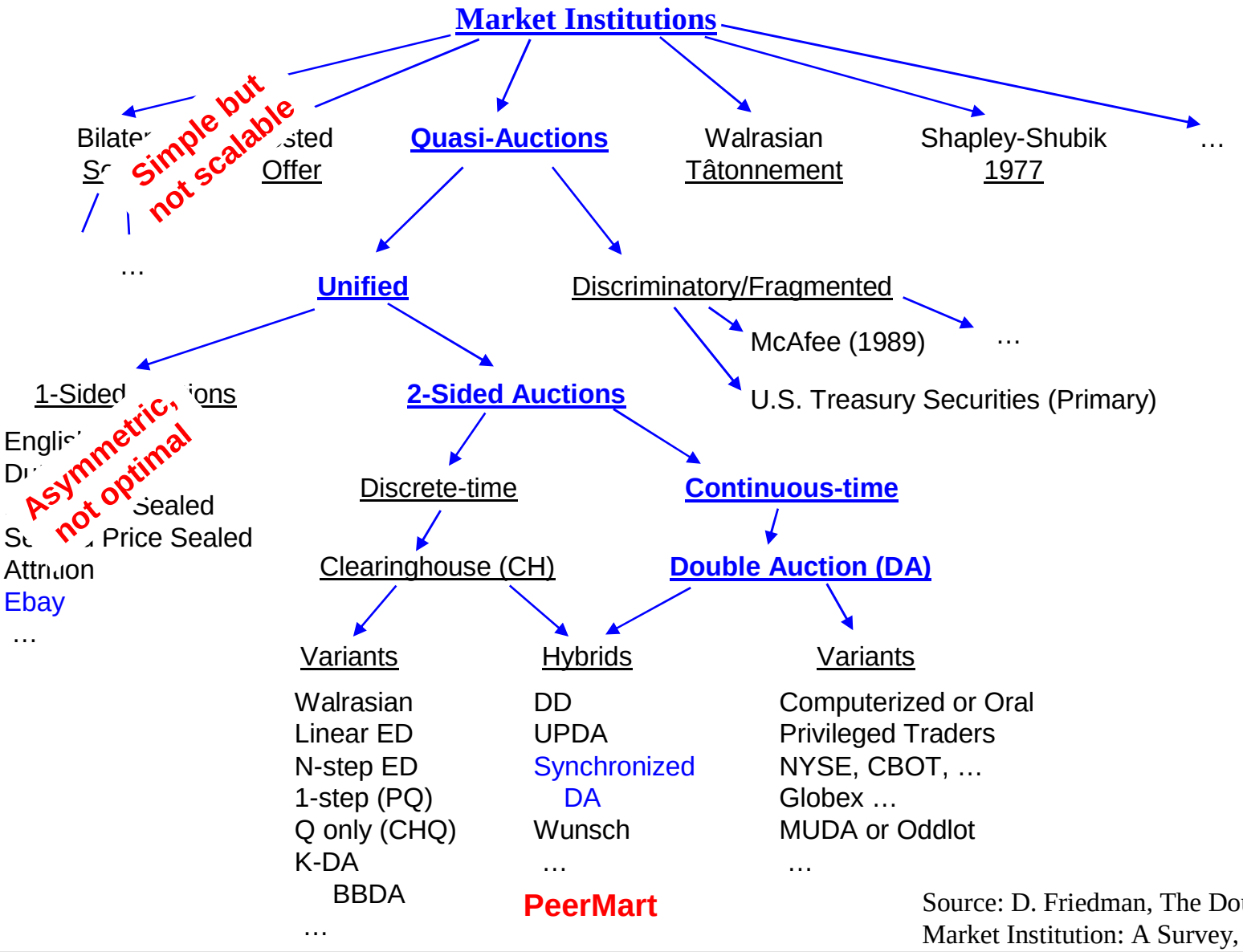
Provider‘s Goal
- Find consumer paying a high price for the service offered



P2P Market: Main Problems

- Why should intermediate peers cooperate?
  - Forwarding messages (requests / bids / offers etc.)
  - Caching / Processing data on behalf of other peers
- Cooperation is costly
  - Intermediate peers have simply no incentives to cooperate
- Even worse: Intermediate peers might be competitors
  - Provider peers could lose opportunity to sell a service
  - Consumer peers could lose opportunity to buy a service
- Peers are autonomous
  - They may act in a selfish or malicious way
  - Currently not considered by most P2P algorithms
- Accounting for forwarding messages etc is not feasible
  - Difficult to measure, high technical effort

P2P Market: Design Space for Pricing



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

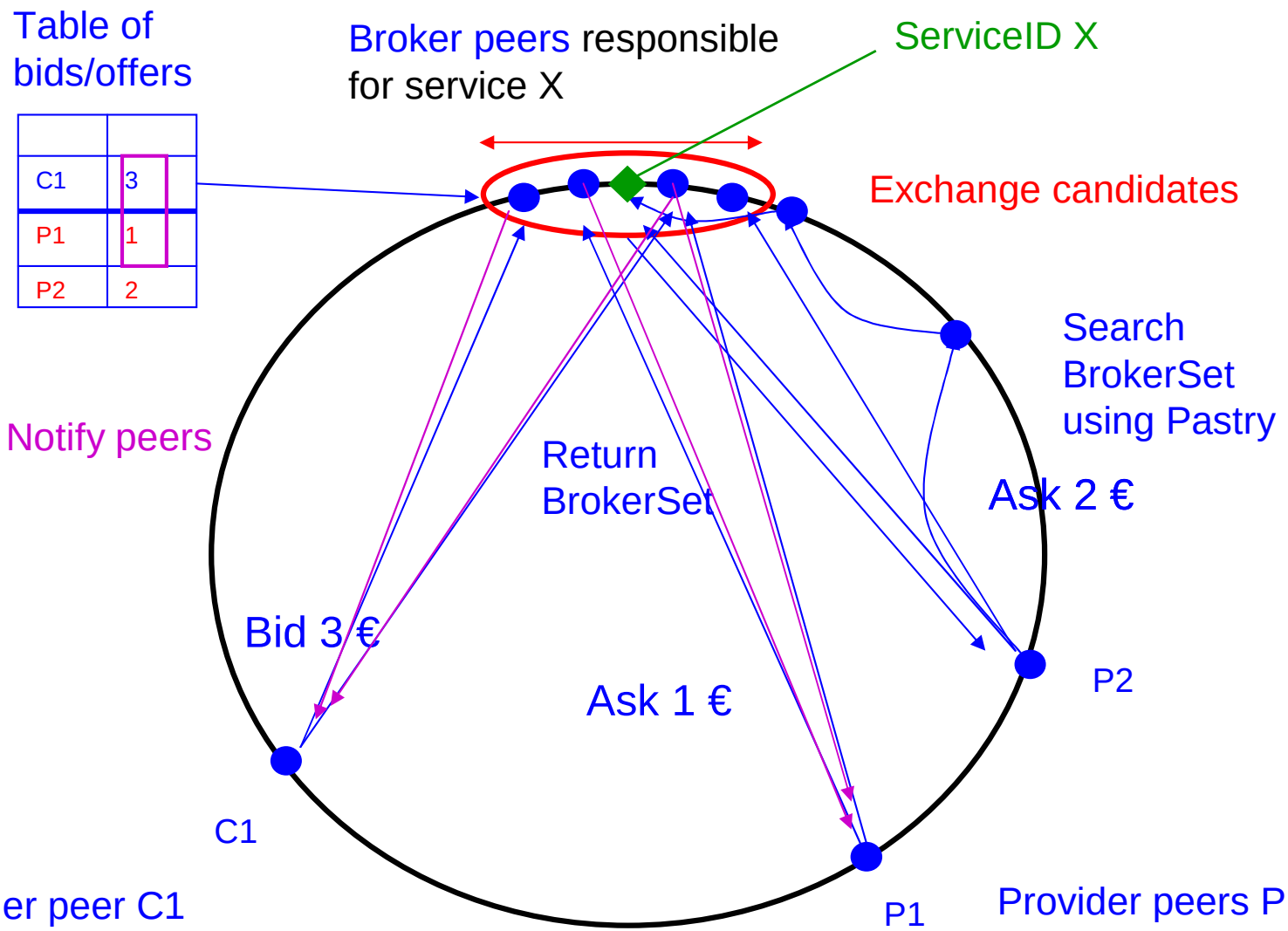
PeerMart's Approach: Distributed Auctions

- Idea
  - Providers offer prices for services, consumers bid for them
  - Offers and bids are optimally matched by broker peers
- 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 offer (bid), if offer (bid) is higher (lower) than current bid price (ask price) => no match, store offer (bid) in a table
    - Otherwise, forward offer (bid) to highest bidder (lowest offerer)

PeerMart's Technical Design

- Pastry is used as underlying P2P infrastructure
  - Open source implementation (FreePastry)
  - Unique 128-bit nodeIds, n closest nodeIds (leaf-set) stored in routing table
- Services are mapped onto a redundant set of broker peers (broker-set)
  - Assumption: Services have unique IDs (e.g. hash value of a file)
  - Redundancy increases reliability and trustworthiness
- Consumers/Providers randomly select f out of n broker peers
  - Load is uniformely distributed on all broker peers
  - n can be increased for highly popular services
- Each broker peer keeps table of m highest bids and m lowest offers
  - Lower bids and higher offers are dropped or rejected
- Matching is performed in a decentralized way
  - Candidates for a match are exchanged between brokers
  - Final matches are determined using majority decisions
  - Consumers/Providers are notified by the f brokers that received the offer/bid

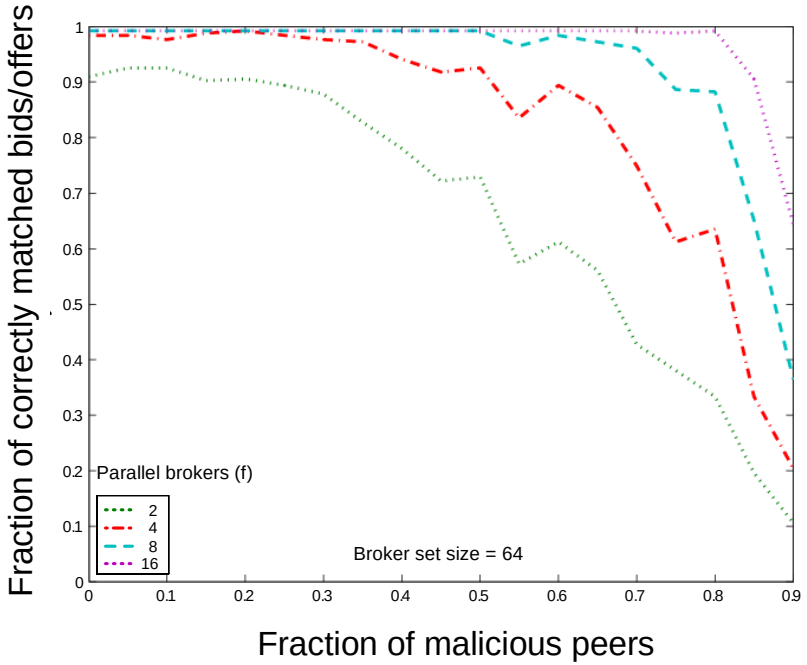
Example: Double Auction in PeerMart



Evaluation

- Complexity Analysis
  - Basic effort for maintaining Pastry: $O(\log_b N)$
  - Finding the responsible broker-set: $O(\log_b N)$
  - All subsequent communication is direct
    - Send offer / bid is O(f)
    - Exchange offers / bids is O(n)
    - Notify providers / consumers is O(1)
- Simulation
  - Assess reliability of PeerMart
  - 500 providers, 500 consumers, 1 service

Reliability



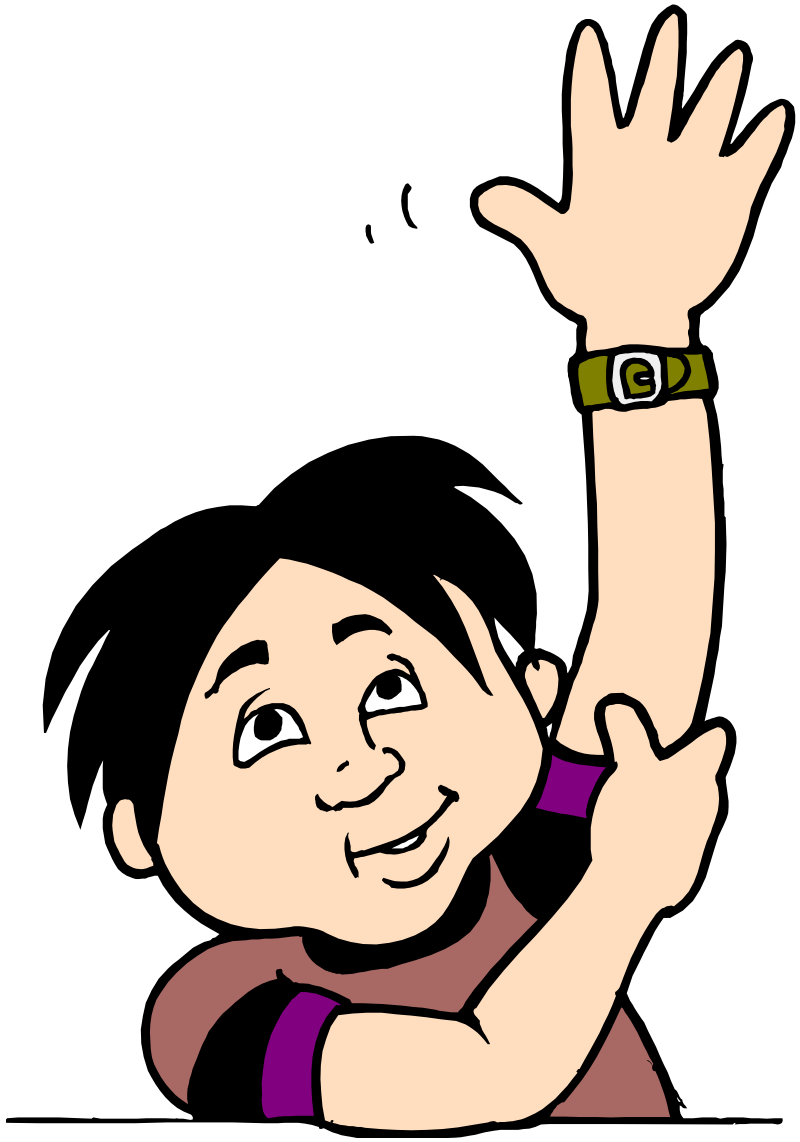
Simulation parameters:
- Bidding strategy for consumers: min (ask_price + 0.1* (bid_limit – ask_price), bid_limit)
- Bidding strategy for providers: max (bid_price - 0.1* (bid_price – offer_limit), offer_limit)
- Reservation prices (bid/offer limits) normally distributed such that ~ 50 % of bids/offers match
- Malicious peers do not forward offer / bid

Conclusion, Status and Further Work

- Mechanism scales well
- High reliability can be achieved through redundancy in presence of malicious peers
- Prototype implemented based on FreePastry
- Run more extensive experiments with prototype
  - Several services with different popularity
- Open Issues
  - Prevent (D)DoS attacks
  - Public key generation and lookup
  - Overlay splitting
  - Punish peers that do not stay to an offered price

    => Use Reputation for Trust

Thank you! – Any Questions?



Backup: Exchange Candidates

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

  => Introduce time slots greater than max Round Trip Time