PSH: A Private and Shared History-based Incentive Mechanism

Thomas Bocek[1], Wang Kun[2], Fabio Victora Hecht[1], David Hausheer[1], and Burkhard Stiller[1,3]

[1] *Department of Informatics IFI, Communication Systems Group CSG, University of Zurich*
[2] *Research Institute of Telecommunications Transmissions, CTTL, China*
[3] *associated with the D-ITET, ETH Zürich*

Motivation and Background

Basic Design, PSH Algorithm

Implementation, Evaluation

Introduction

- P2P Systems have many advantages over centralized approaches
  - Scalability, load balancing, redundancy, no SPOF, ...
  - Popular applications: Skype, BitTorrent, Zattoo, Emule, ...

Problem Statement

- Problems in decentralized network
  - Free-riding, selfish behavior, malicious peers
    - Gnutella: 70% peers share no files
  - → Incentives schemes necessary!
    - Tit-for-Tat (TFT), Private Shared History Incentive Mechanism (PSH)
- TFT incentive mechanism for file-sharing (BitTorrent)
  - TFT incentive mechanism works well with symmetric interest
    - A downloads from B, B downloads from A chunks from same file
    - TFT incentive mechanism uses private history (aggregated information from local observation)
- TFT incentive mechanism enables fair sharing
  - Provide unused resources now, demand resources later

Motivation

- TFT fails for asymmetric resource interest
  - Example DHT: Peer A stores data on B, but B does not store anything on A
  - Example IPTV: Channel switching in IPTV, new channel may require building up private history

→ For asymmetric resource interest: Transitive TFT
  - Peer A downloads from B, B downloads from C, C downloads from A
  - Key question: How does C know about A and vice versa
    - Use shared history (observations from other peers)

Basic Design

- **P**rivate **S**hared **H**istory Incentive Mechanism (PSH)

- PSH combines private and shared histories

  1st step: Collect information using shared history

  2nd step: Evaluate and verify information using private history

Background

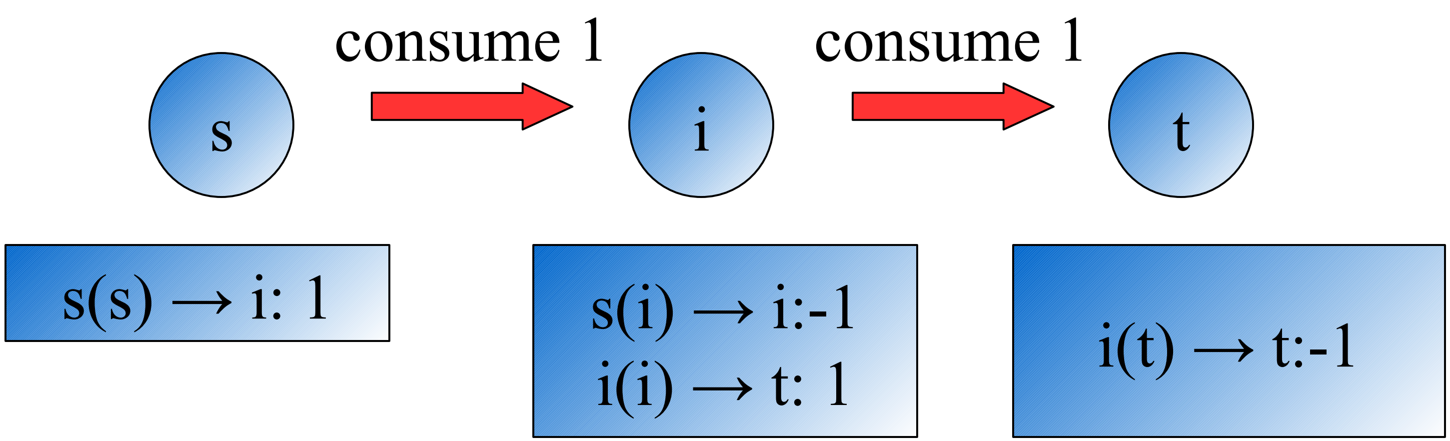
- Private history (TFT)
  - Advantages
    - Collusion resistant
    - No transaction information distribution overhead
  - Disadvantages
    - Works with symmetry of interest
- Shared history (Transitive TFT)
  - Advantages
    - Works with asymmetry of interest
  - Disadvantages
    - Not collusion resistant / not performing well
    - Transaction information distribution overhead

PSH Advantages / Disadvantages

- Private history (TFT)
  - Advantages
    - Collusion resistant
    - No transaction information distribution overhead
  - Disadvantages
    - Works with symmetry of interest
- Shared history (Transitive TFT)
  - Advantages
    - Works with asymmetry of interest
  - Disadvantages
    - Not collusion resistant / not performing well
    - Transaction information distribution overhead

PSH Algorithm (1)

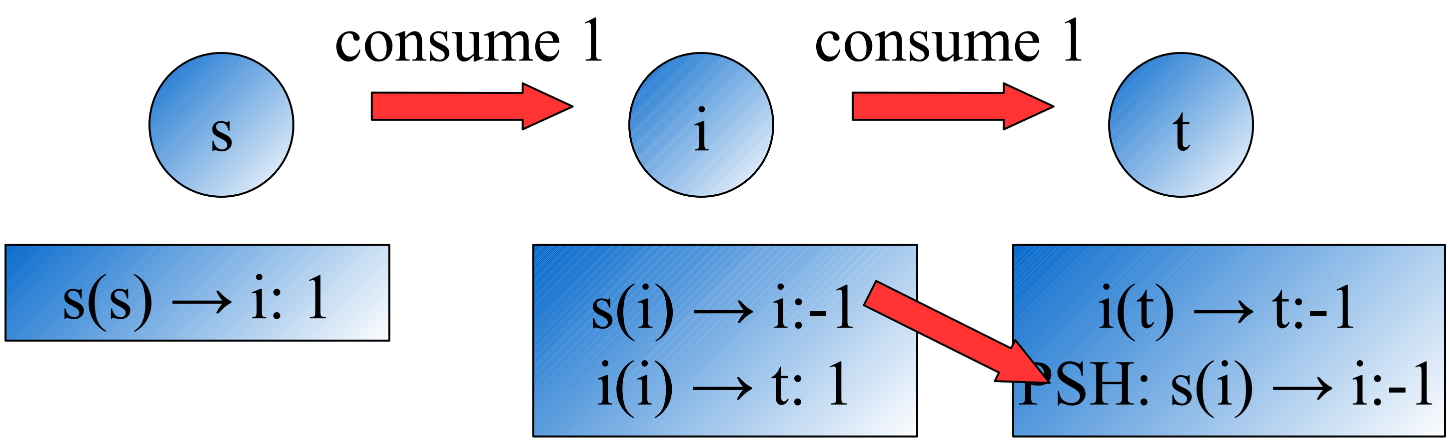
- PSH algorithm in 4 steps with examples, initial credit 1
  1. Distribute latest transaction information with every request/reply
     + no new connection required, - message size larger



Notation *v(w)→x:z* peer v consumes z resources from peer x, observed by peer w

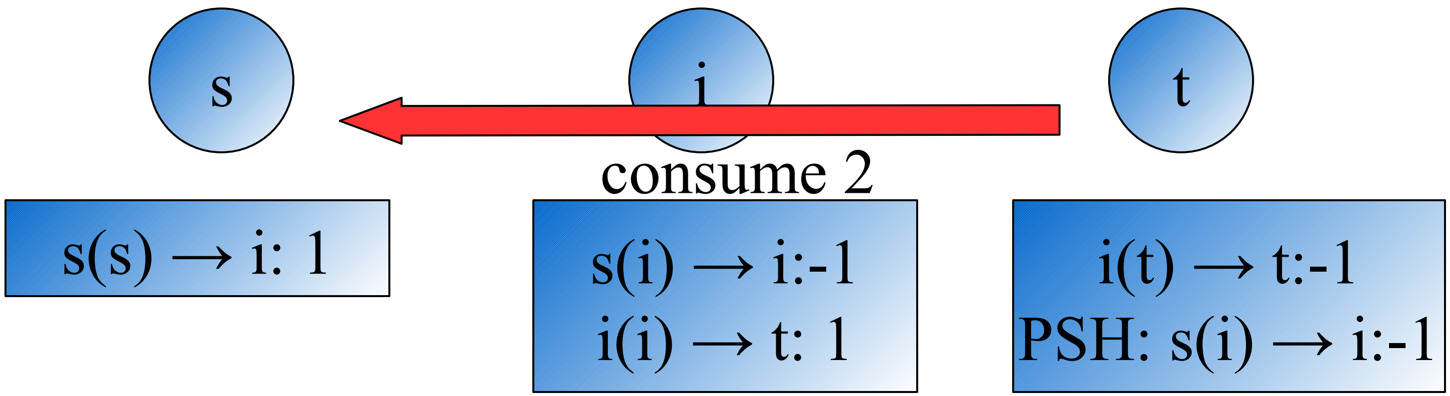
PSH Algorithm (1)

- PSH algorithm in 4 steps with examples, initial credit 1
  1. Distribute latest transaction information with every request/reply
     + no new connection required, - message size larger



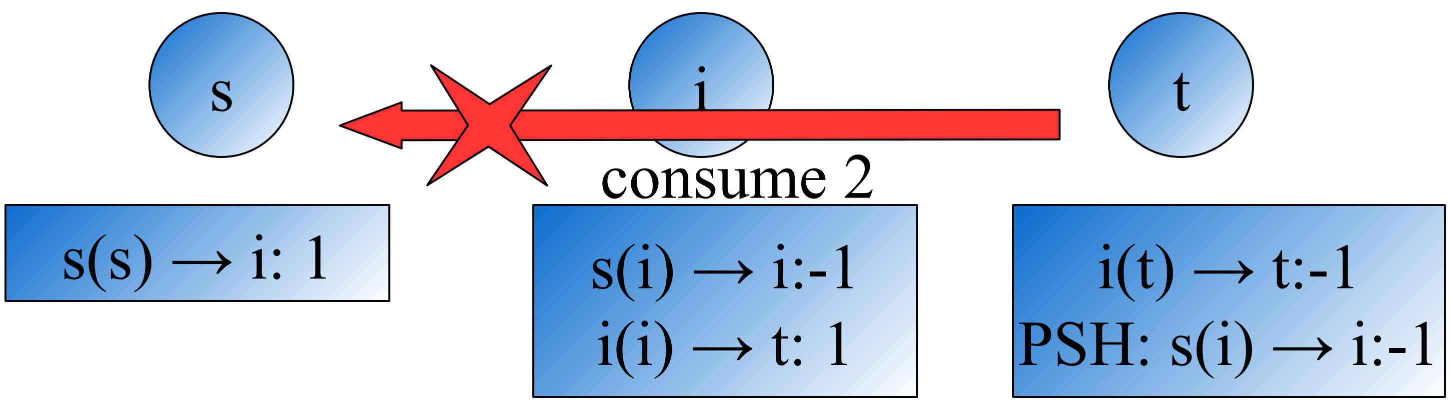
PSH Algorithm (2)

2. If TFT fails for transaction t, path of length l>2 is searched in shared history



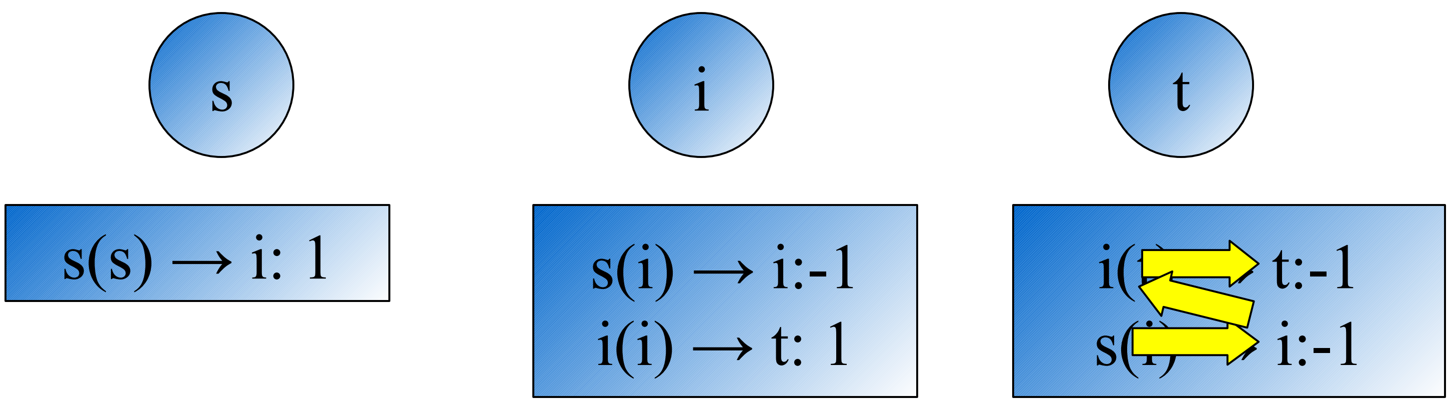
PSH Algorithm (2)

2. If TFT fails for transaction t, path of length l>2 is searched in shared history



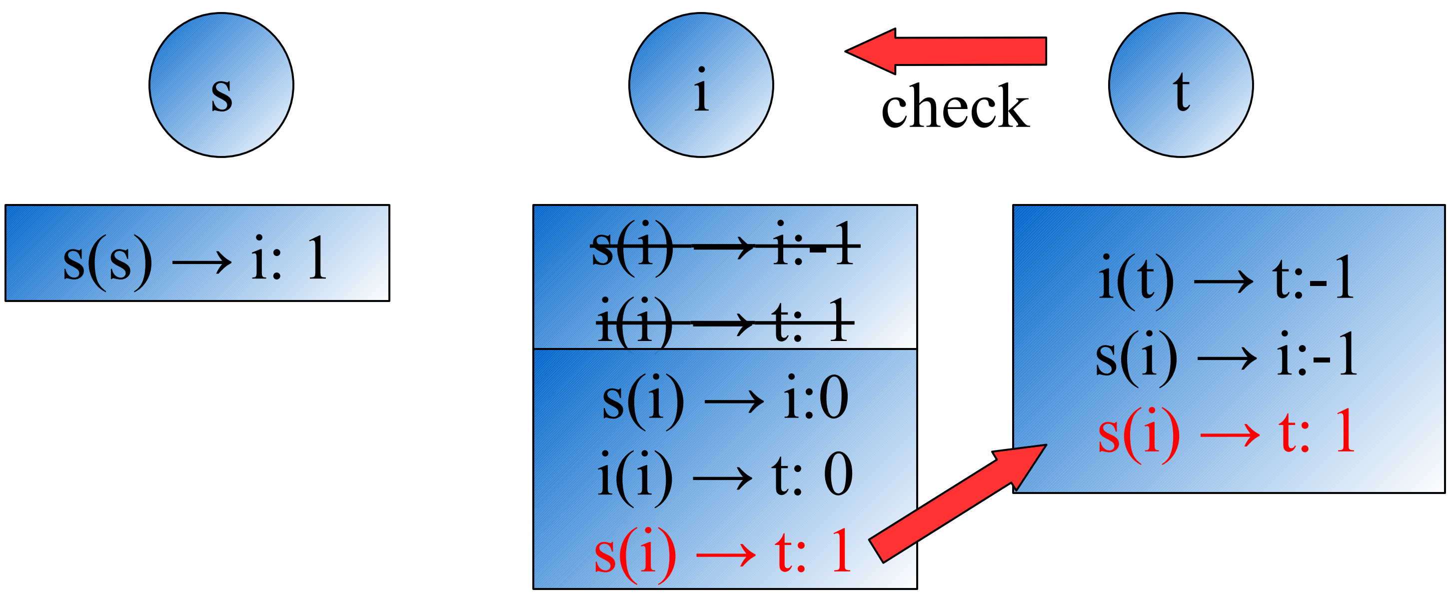
PSH Algorithm (3)

2. If TFT fails for transaction t, path of length l>2 is searched in shared history



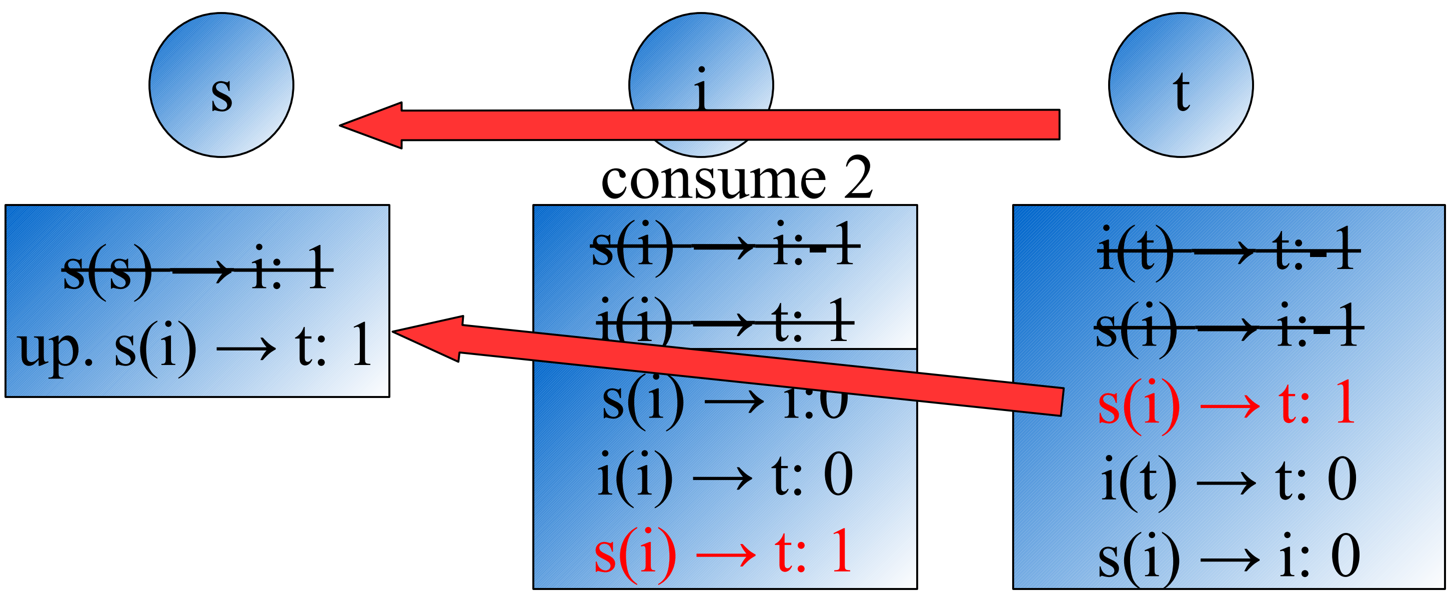
PSH Algorithm (4)

3. If path is found, issue check c
4. Apply check c on peers involved in transaction t and execute transaction t again



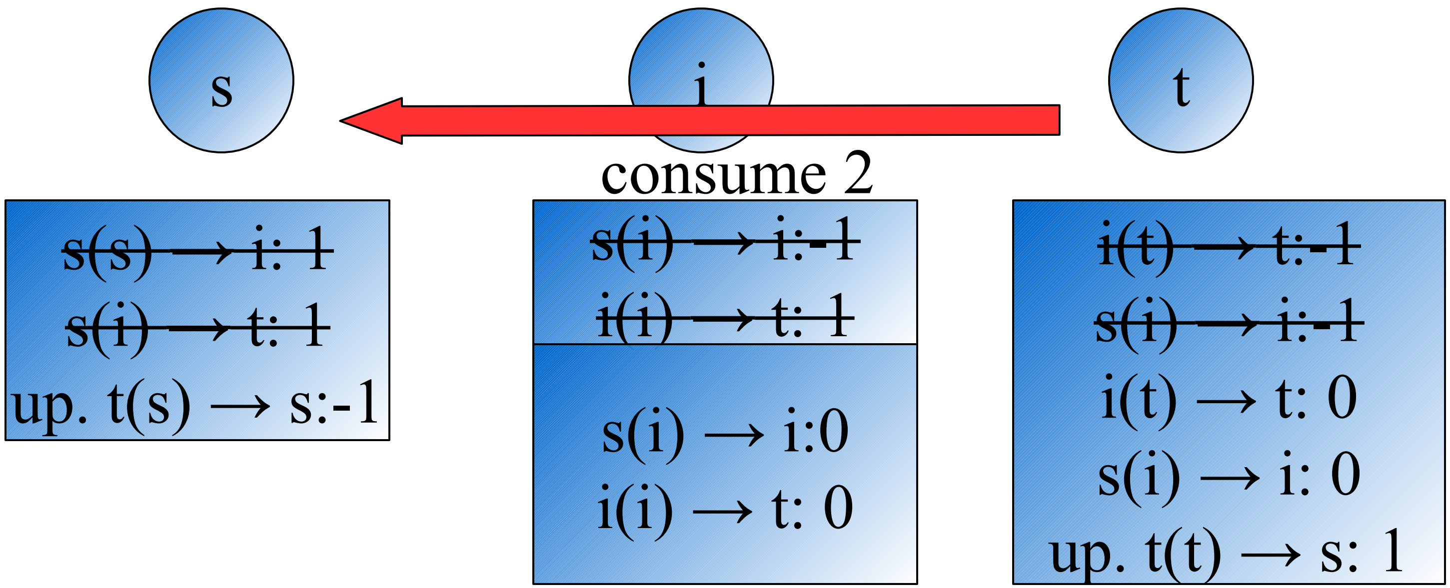
PSH Algorithm (4)

3. If path is found, issue check c
4. Apply check c on peers involved in transaction t and execute transaction t again



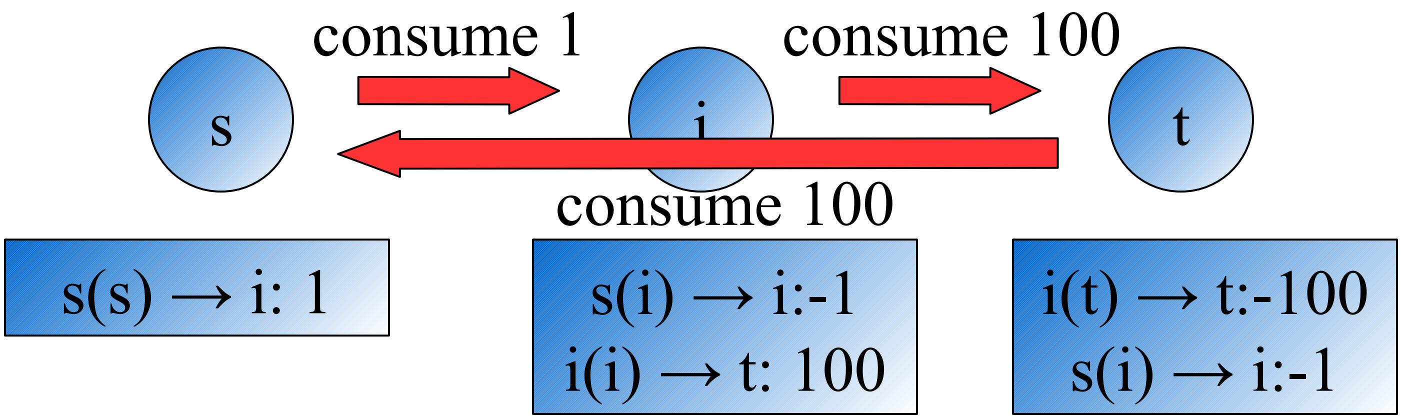
PSH Algorithm (4)

3. If path is found, issue check c
4. Apply check c on peers involved in transaction t and execute transaction t again



PSH Algorithm – Collusion-proof

- Maximum flow for path s → i → t is 1



- Missing updates/wrong checks/outdated information/ malicious peers
  - Peer s decides based on its private history
  - Balance on peer i remains the same
  - For peer t: amount on check may be lost

PSH Algorithm - PSH_r

- PSH sends with each request latest transaction information
- PSH_r reduces distribution overhead
  - – No distribution of transaction information beforehand
  - – If transaction fails, transaction information exchanged

    + less overhead

    + smaller messages

    - only paths of length 3 can be found reliably (one intermediate node)

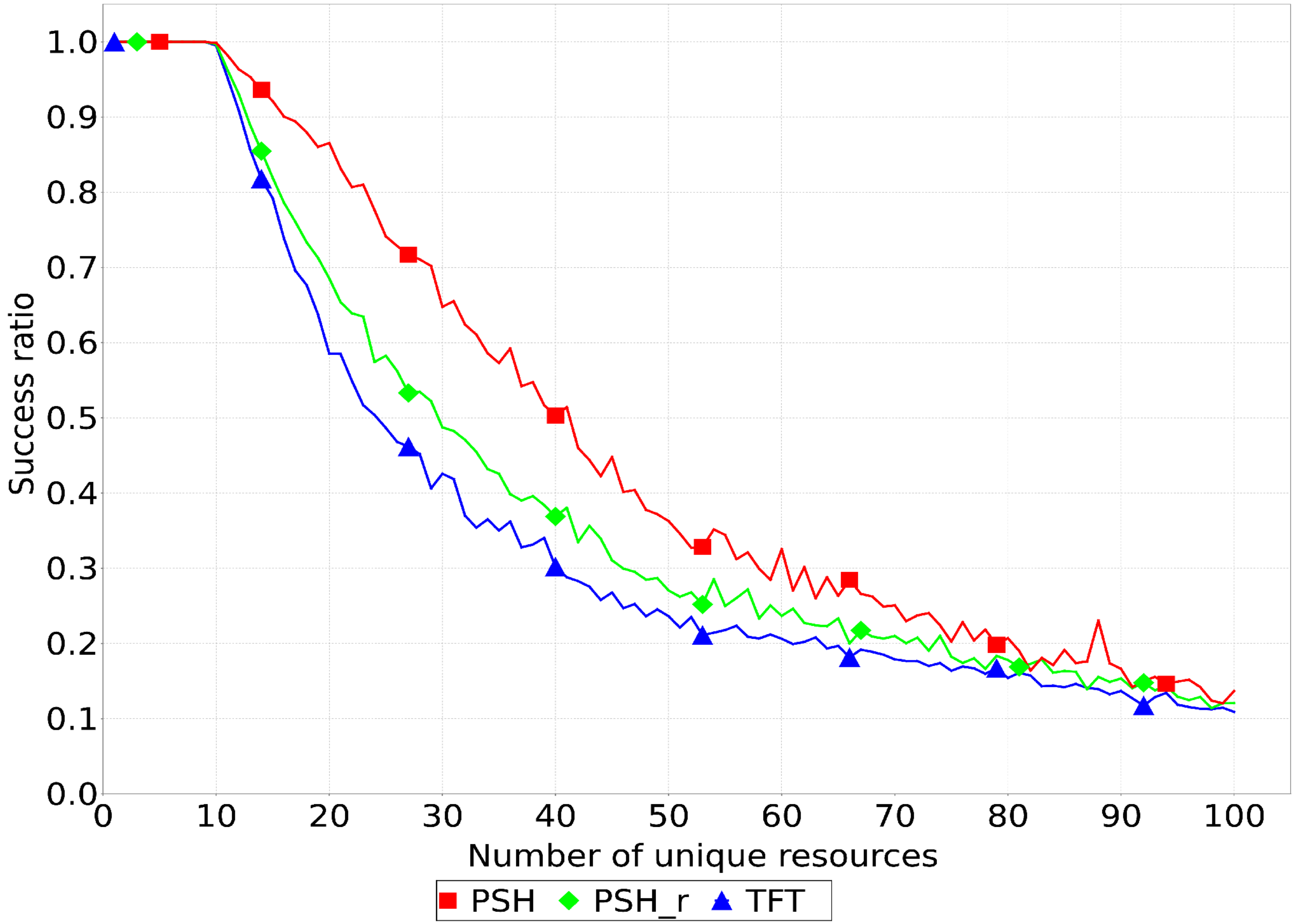
Implementation

- PSH implemented in Java
- PSH uses a DHT based on Kademlia
- Evaluated with n peers, n=[1..N], N=100
- Experiments run on one machine, operation executed sequentially
- Compared mechanisms:
  - PSH (max. 3 retries)
  - PSH_r (no retries)
  - TFT (no retries)

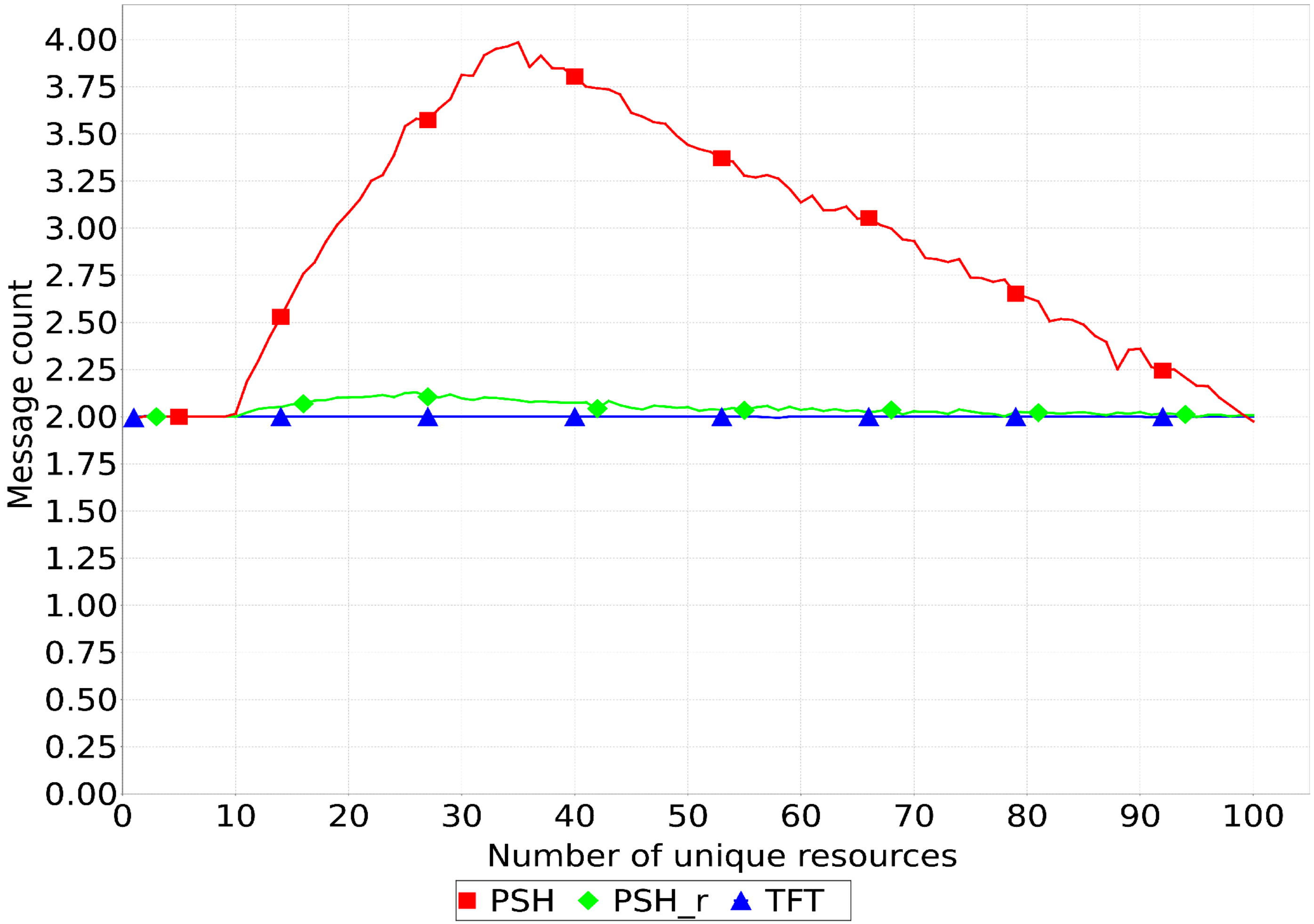
Evaluation (1)

- Parameter "number of unique resources" varied, p=[1..N], N/p peers share the same resource
  - p=1 → all peers shares the same resource (symmetric)
  - p=100 → all peers share different resource (asymmetric)
  - p=50 → 2 peers share same resource
- Plotted against:
  - Success ratio
    - With perfect knowledge success ratio is 1.0
  - Average message count
    - Number of messages per resource request/reply
  - Message size
    - Total size for all messages

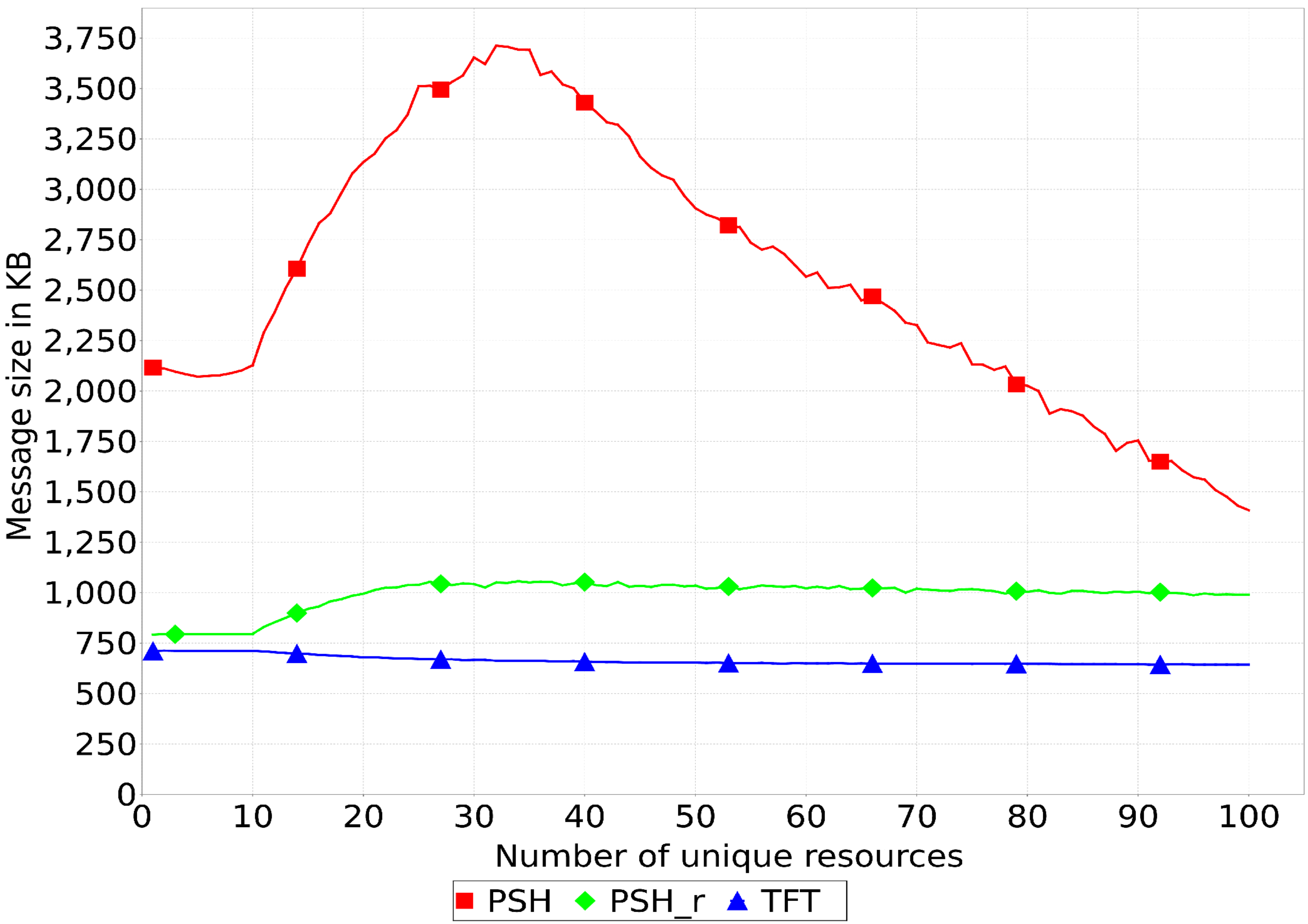
Evaluation (2) – Success Ratio



Evaluation (3) – Count



Evaluation (4) – Size



Summary and Conclusions

- PSH combines private and shared history
- PSH works up to 73% better than TFT incentive mechanism for asymmetric resource interest
- Distribution of transaction information introduces overhead

Future Work

- Reduce message size: WIP PSH2

- Distributed testing environments
  - Deployment on EmanicsLab

- Possible PSH applications
  - P2P TV (channel zapping)
  - P2P Voting

Questions?

Thank you for your attention.

Reference:

T. Bocek, W. Kun, F. V. Hecht, D. Hausheer, B. Stiller: *PSH: A Private and Shared History based Incentive Mechanism.* 2nd International Conference on Autonomous Infrastructure, Management and Security Resilient Networks and Services (AIMS), July 2008