The Design of A Distributed Rating Scheme for Peer-to-peer Systems

Debojyoti Dutta\textsuperscript{1}, Ashish Goel\textsuperscript{2}, Ramesh Govindan\textsuperscript{1}, Hui Zhang\textsuperscript{1}

\textsuperscript{1}University of Southern California
\textsuperscript{2}Stanford University
Outline

- Research motivations
- Basic design issues in P2P rating schemes
- A distributed rating scheme to incentivize cooperation in P2P file-sharing systems
- Dealing with collusion and malice
- Conclusion & future work
Research motivations

- Object: P2P file-sharing systems
  - Open social communities.
  - An explicit reputation layer was ignored in the original design.

- Goal: Build reputation in such systems
  - Incentive for user participation
    - free-riding phenomenon [Adar et al. 2000] [Saroiu et al. 2001]
  - Isolation of malicious users
    - distribution of inauthentic files
    - propagation of virus or worms [VBS.Gnutella] [Fizzer.Kazza]
**P2P rating: basic design issues**

- “Distributed” rating
  - following P2P design philosophy.

- “Efficient” rating
  - low cost to run and maintain this reputation system.

- “Collusion-proof” rating
  - Effectiveness.
A distributed rating scheme

- To incentivize cooperation in P2P file-sharing systems

- Main components
  - Positive rating
  - Rating verification scheme
Positive rating

• The recognized service done to the community
  
  • $R_i$ of user $i$: non-decreasing with the number of successful requests that it has satisfied within some sliding time window.
  
  • The higher $R_i$ user $i$ has, the better service it gets from the network.
Verification-based rating scheme

Rating $R_i$

Data request

(i, $R_j$)

User i

Verify if $R'_i$ is true for i

User j
Two verification schemes

- Structured verification scheme (SVS)
  - Each user has a set of designated supervisors which keep its up-to-date reputation information.
  - The supervisors are responsible for the verification.

- Unstructured verification scheme (UVS)
  - A user j queries some of user i’s claimed customers for the verification, and believes i when the majority of the probed users reply with a “yes”.

6/6/2003  P2Pecon  8
Assumption

- Users are distinguished by their IP addresses.
  - At a given time, one IP address corresponds to an unique user.
SVS – the supervising topology

• In the supervisory directed graph
  ❑ Any user is random to its supervisors.
  ❑ No small supervising loop exists.
  ❑ There is a fast reactive approach for any user $j$ to deliver a message to any other user $i$’s supervisors, and the path never includes $i$. 
A Chord[stoica2001] supervising overlay

A Chord network with 8 users and 8-bit key space

6/6/2003
P2Pecon 11
SVS – the supervising topology

- In the supervisory directed graph
  - Any user is random to its supervisors.
  - No small supervising loop exists.
  - There is a fast *reactive* approach for any user j to deliver a message to any other user i’s supervisors, and the path never includes i.
Rating verification in SVS

Yes/No

user i’s supervisors

ID(i)

verification request

user j

Supervising overlay

user i
Structured verification scheme

- “Distributed” rating ✓
- “Collusion-proof” rating ✓
- “Efficient” rating ?
  - Extra cost to maintain a supervisory overlay when the underlying network is not DHT-based.
  - Repetitive actions when there are multiple supervisors.
Unstructured verification scheme

1. When user j decides to verify user i’s rating, it gets a portion of i’s customer list, and asks the users on the list if i did the claimed service to them.

- The customer samples should be random on the full customer list of node i.
- Disclosure of full customer list could raise privacy concern, and incur high communication cost.

2. When the majority of the probed users reply with a “yes”, j is convinced that $R'_i$ is $R_i$. 
Randomly sampling without the complete customer list

user i’s customer list

\[
\begin{array}{cccccccc}
\text{a} & \text{b} & \text{c} & \text{d} & \text{e} & \text{f} & \text{g} & \text{h} \\
\end{array}
\]

hashing

user i’s customer vector

\[ [2, 3, 1, 2] \]
P2P users

users

selfish  malicious  altruistic

non-colluding  colluding
Colluding selfish users

• Possible solution 1: discrete rating
  - Grade★ : if a user has served no more than 10 users.
  - Grade★ ★ : if a user has served between 10 and 100 users.
  - Grade★ ★ ★ : if a user has served more than 1000 users.

• Possible solution 2: rating as virtual currency
  - A user has to pay (reduce its rating) for the service it claims to have received.
  - SVS: asks the requestor’s supervisors for a payment.
  - LUVS: future work.
Colluding malicious users

- One possible strategy
  1. A user $i$ quickly earns a high rating by faking transactions with other colluding users.
  2. User $i$ then does bad things until earns bad-enough reputation.
  3. User $i$ quits the network to clear its history,
  4. User $i$ rejoins the network and repeats the above actions.
**Conclusion & future work**

- A simple distributed rating scheme to incentivize cooperation in P2P file-sharing systems.
  - Two distinct verification schemes

- Refine LUVS scheme to handle colluding selfish users.

- Refine our rating scheme to be collusion-proof.