Making P2P Downloading Dependable by Exploiting Social Networks

Yingwu Zhu

Seattle University

Outline

- Background
- Anti-pollution solutions
- System design
- Experimental setup & results
- Conclusions

Background

- Significant file pollution in P2P file sharing networks (e.g., INFOCOM'05, IPTPS'06)
- Pollution sources
  - Users accidentally create damaged files and upload them
  - Attackers and companies (e.g., Overpeer) intentionally deposit a massive amount of decoys
  - Users download polluted files and help spread them wider

Background

- Significant file pollution in P2P file sharing networks (e.g., INFOCOM'05)
- Pollution sources
  - Users accidentally create damaged files and upload them
  - Attackers and companies (e.g., Overpeer) intentionally deposit a massive amount of decoys
  - Users download polluted files and further spread them wider

P2P download is No Longer Dependable!

Anti-pollution Solutions (I)

- Heuristic selection
  - Download decisions based on random selection or popularity of replicas in query responses
  - Cons: susceptible to manipulation

Anti-pollution Solutions (I)

- Heuristic selection
  - Download decisions based on random selection or popularity of replicas in query responses
  - Cons: susceptible to manipulation
- Reputation-based
  - File authenticity = peer reputation
  - Requires a reputation system (e.g. Credence)
  - Cons: identity whitewashing, bootstrapping of new peers, Sybil attacks

Anti-pollution Solutions (II)

- Trusted, centralized database
  - Maintains fingerprints of authentic files
  - Database queries provide recommendations
  - Cons: maintenance cost, target of attacks, single point of failures

Our Solution

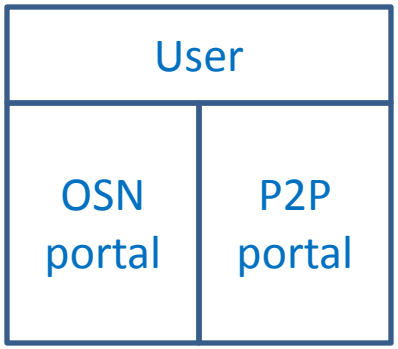
- Builds a recommender atop an OSN
  - Maintained by participating users collectively
- Exploits trust relationships embedded in OSNs
  - Assume public/private keys per user
  - Network links reflect strong social connections
  - Attackers could create many identities, but are limited to create social links to honest users (attack edges)

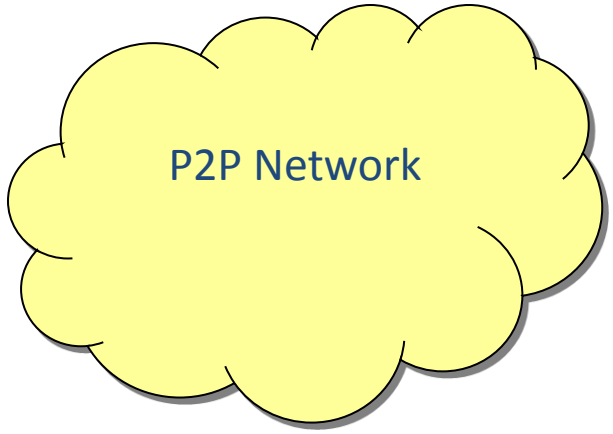
Our Solution

- Exploits trust relationships embedded in OSNs
  - Assume public/private keys per user
  - Network links reflect strong social connections
  - Attackers could create many identities, but are limited to create social links to honest users (attack edges)
- Builds a recommender atop an OSN
  - Participating users collaboratively maintain it

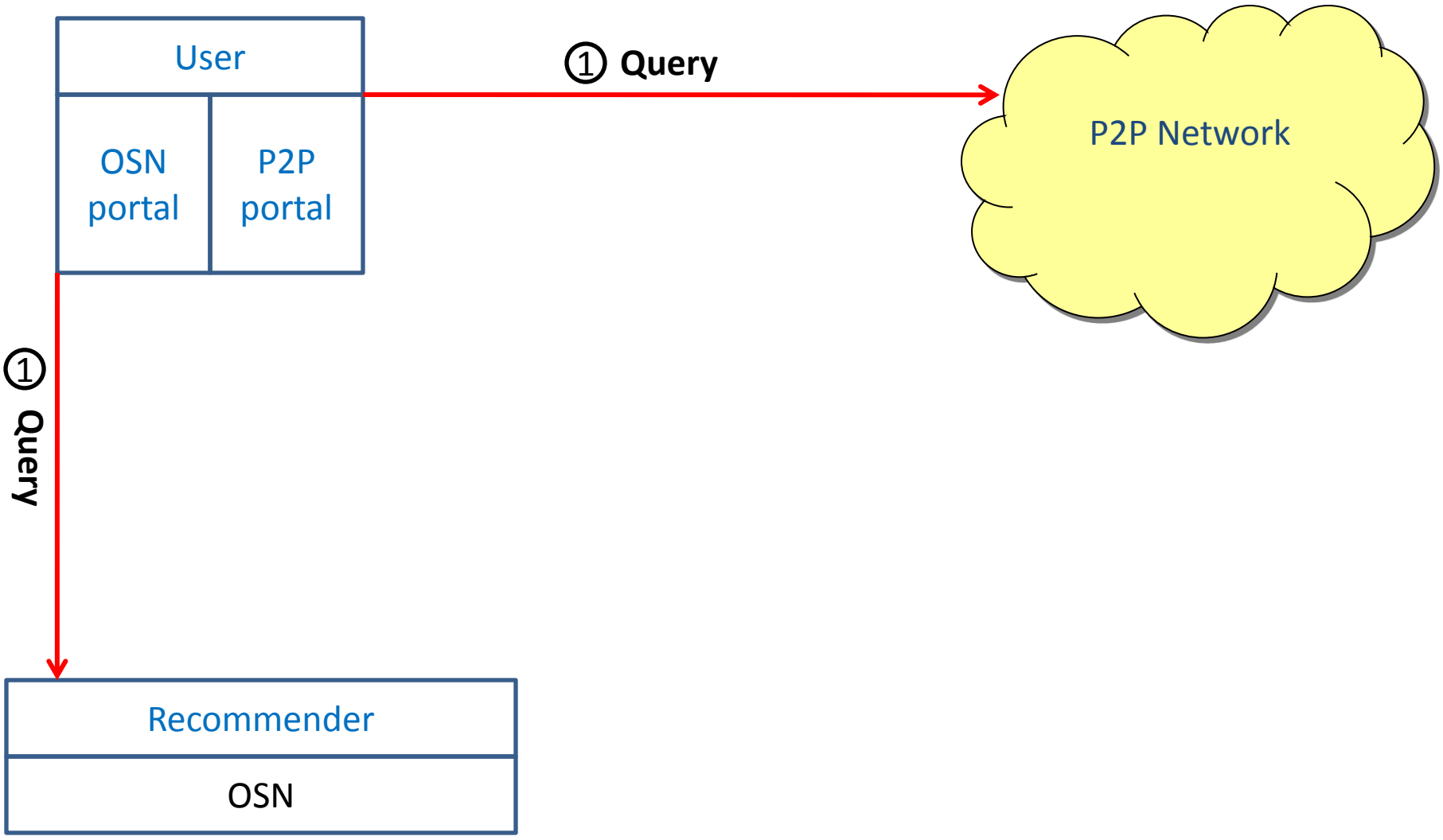
Goal: Make P2P Download Dependable Using Our Recommender!

System Overview

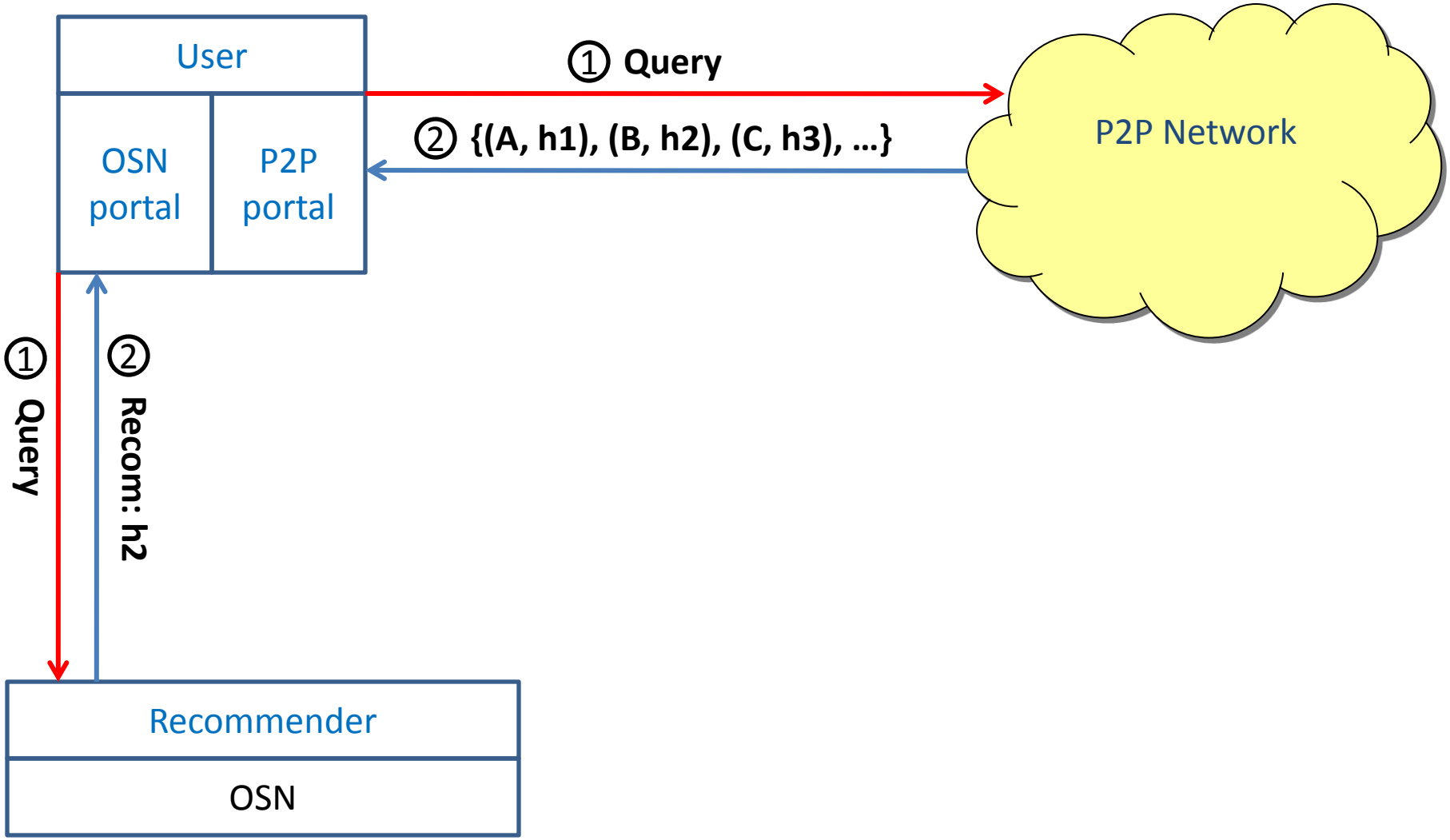




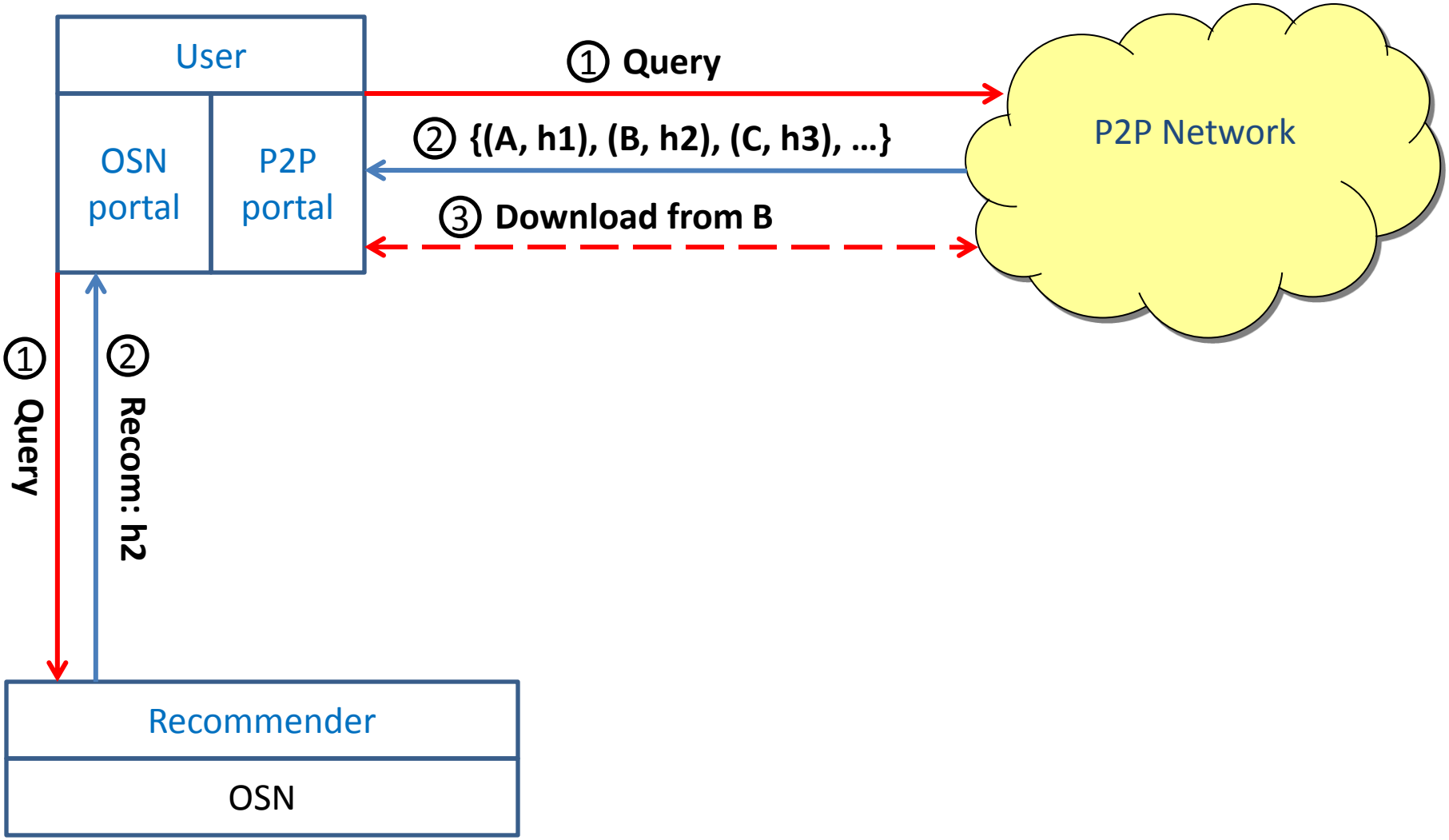
System Overview



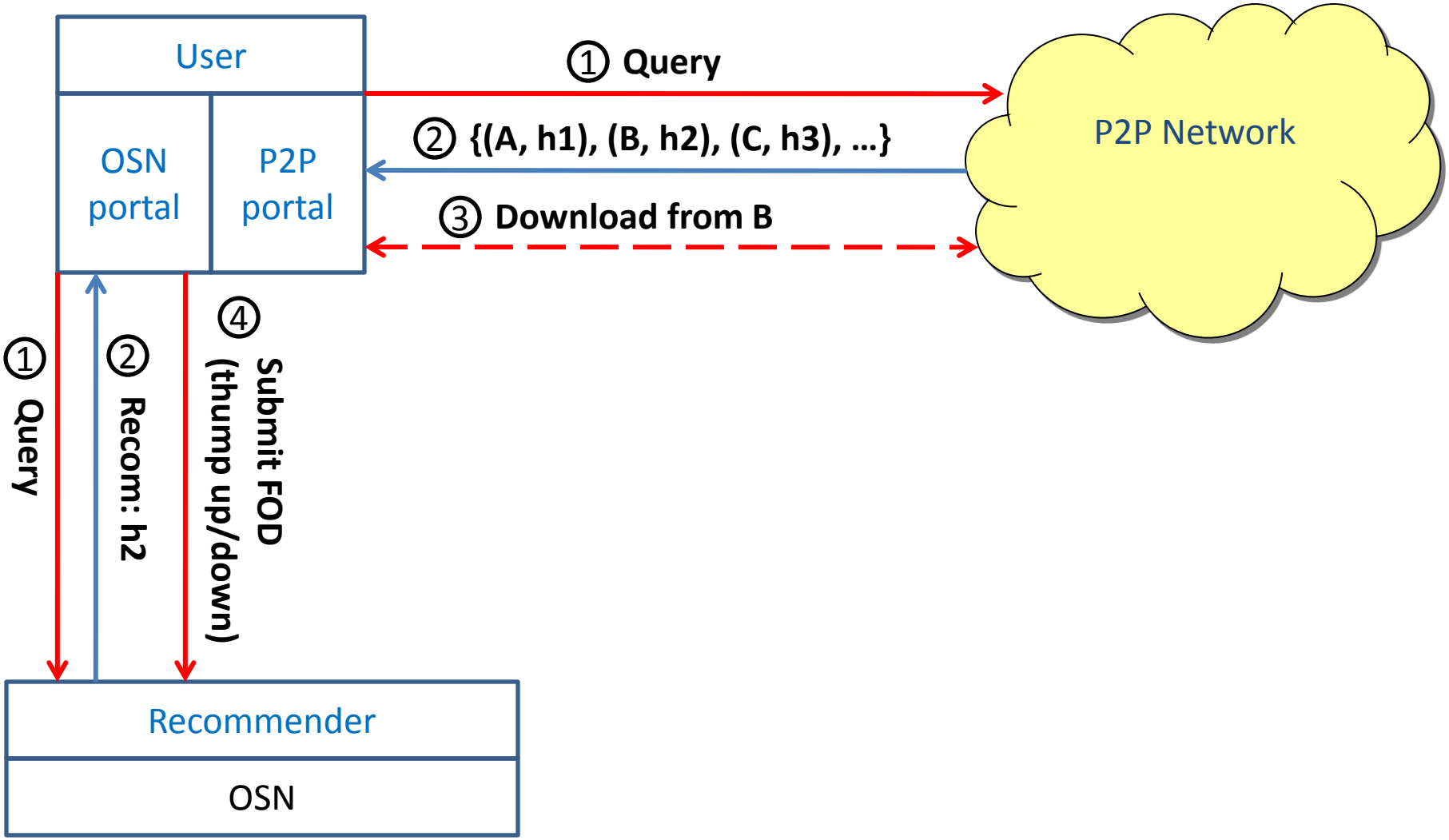
System Overview



System Overview

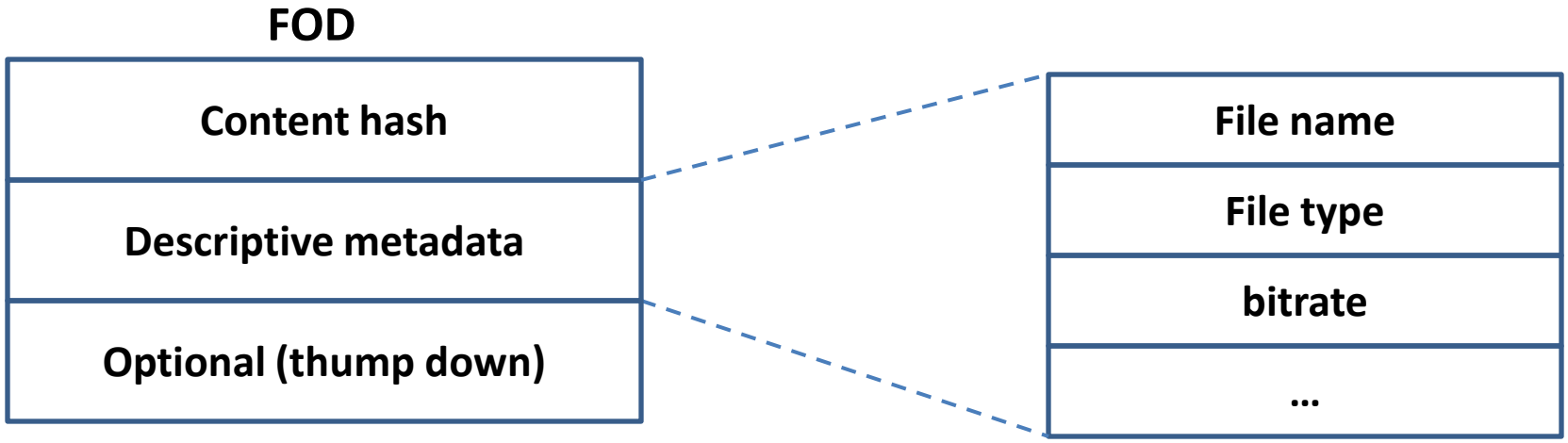


System Overview



Formatted Object Descriptor (FOD)

Maintained in the recommender



1. **Descriptive metadata used to determine whether two FODs describe the same object**
2. **Descriptive congruence: two FODs describe the same objects regardless of contained content hash**

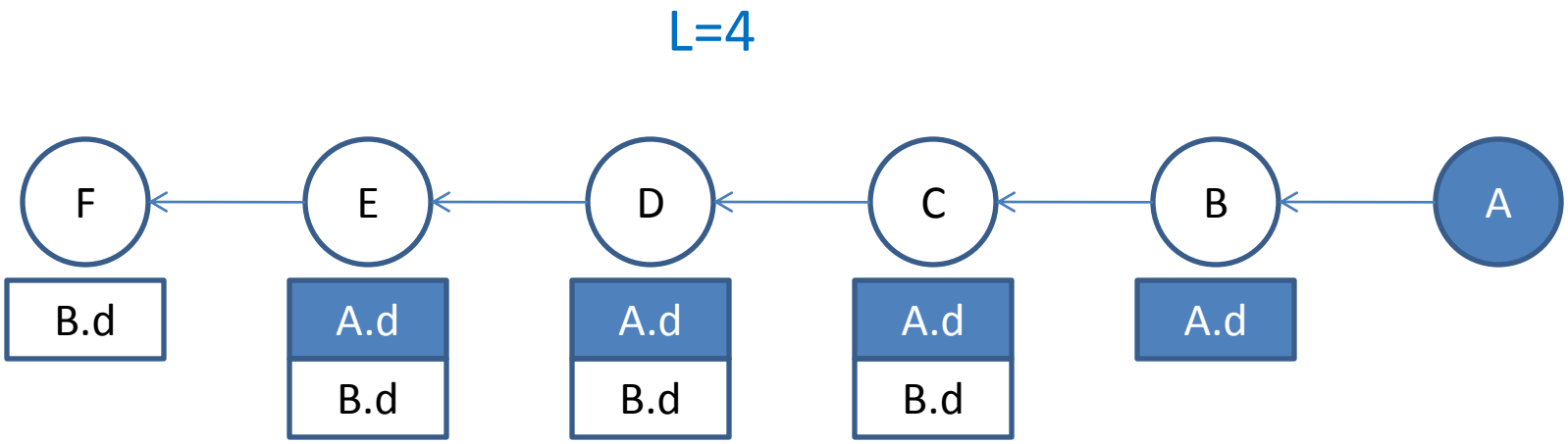
Interfaces

- put (FOD d)
  - To publish a FOD d to the recommender via social links
- h = get (Query q)
  - To return a recommended content hash of the target file indicated by q

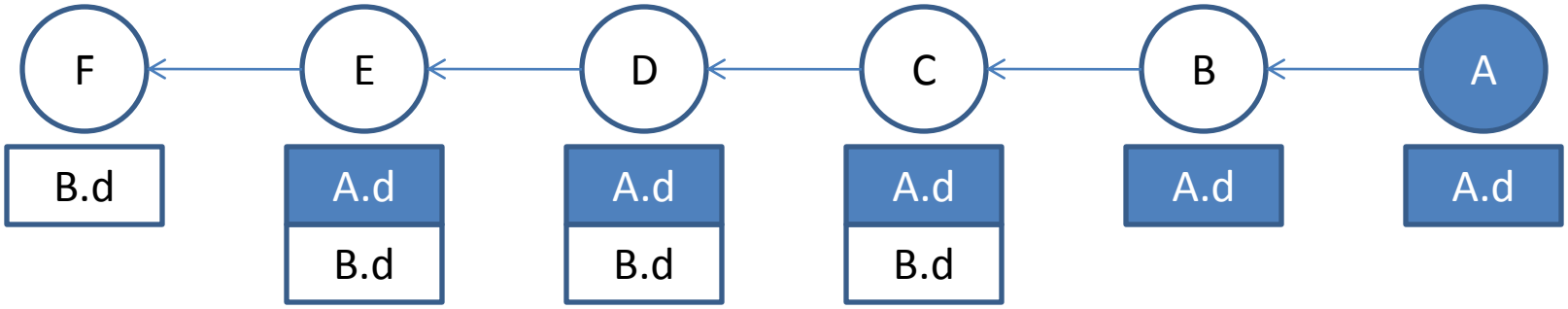
Put FODs (I)

- Uses random routes of length L
  - Random routes ≠ random walks
  - A node w/ d edges: x1, x2, …, xd. Permutation of the edges y1, y2, …, yd. Enter via xi → exit via yi
  - Deterministic & path convergence
  - Limit power of attackers to # of attack edges (m)
  - L controls contamination scope of malicious FODs
  - At most L×m nodes are polluted

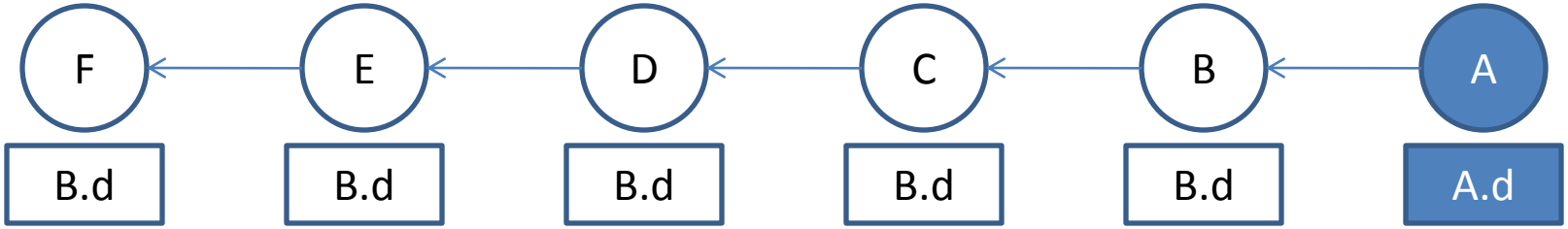
Put FODs (II)



Put FODs (II)



- Overriding policy to further limit pollution of malicious FODs
  - A downstream FOD overrides a upstream FOD if they are descriptive congruence



w/ overriding policy (L = 4)

Get Recommendation

- Duplicate a query over its m social links to aggregate <hash, publisher's public key>
- Ranking: the hash with most unique publishers is returned as recommendation
- Rationale: The random routes of query initiator are more likely to intersect with those of honest nodes than those of attackers due to their limited # of attack edges

Attacks against the Recommender

- Publish malicious FODs
- Drop "put" messages of authentic FODs
- Manipulate routing query (response) "get" messages
- Abuse feedback (by thumb-down FODs) on recommendations
- Whitewashing
  - Malicious users will be blacklisted once identified!

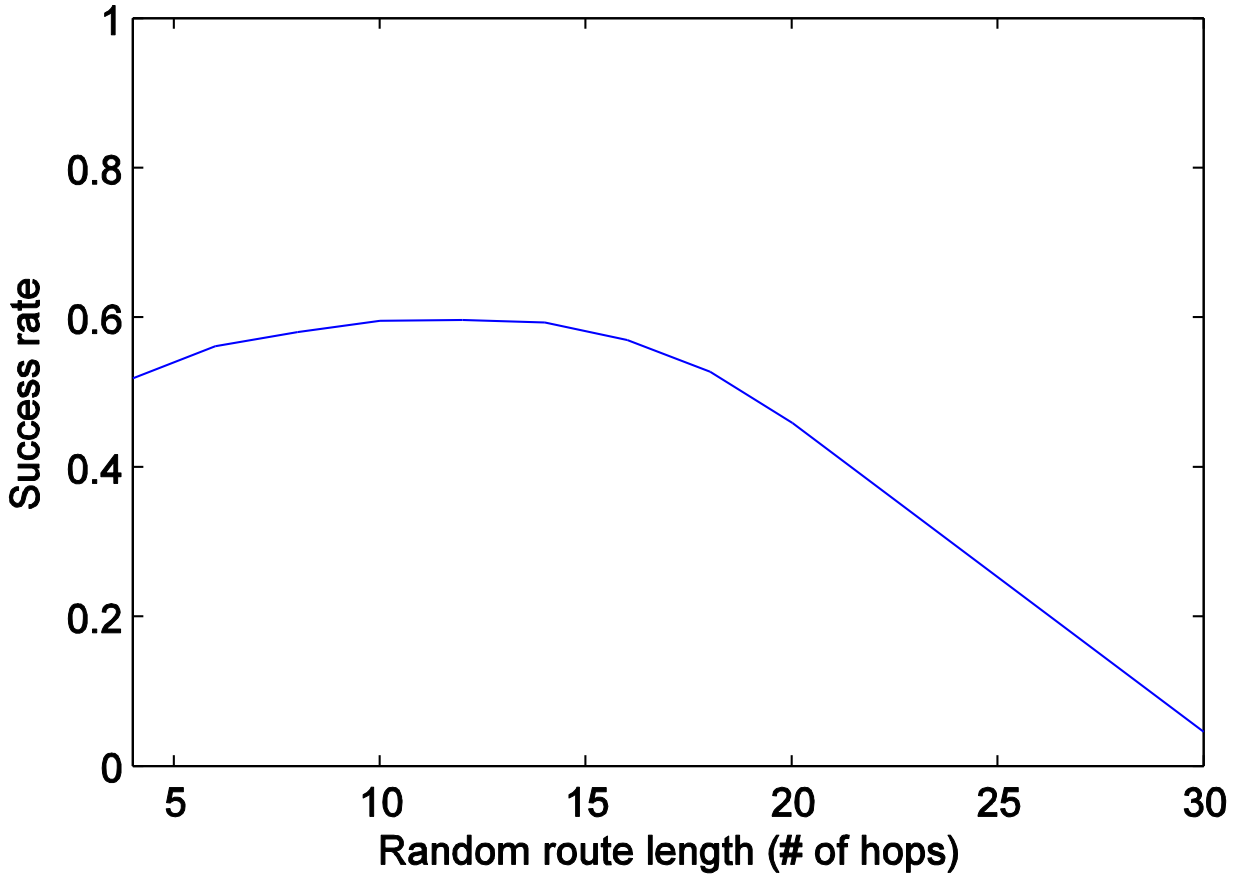
Experimental Setup

Parameter	Value
# of nodes in OSN	10,000
Avg. node degree in OSN	20
OSN topology	Kleinberg's synthetic social network model
Attackers	Start from a seed node BFS to choose attackers until # of attack edges =108

Metric

- ***Query success rate***: fraction of queries that get right recommendation

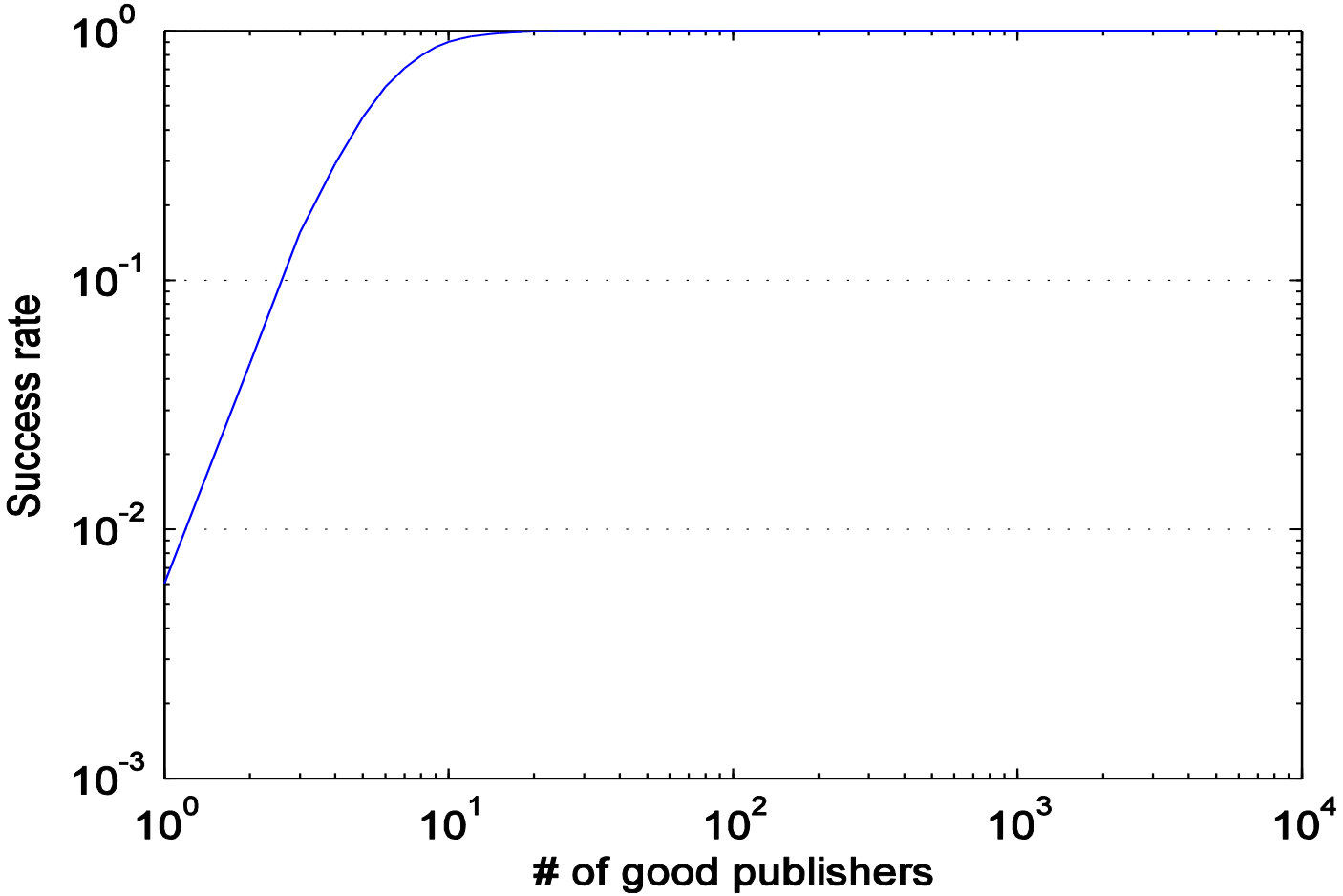
Query success rate vs. random route length



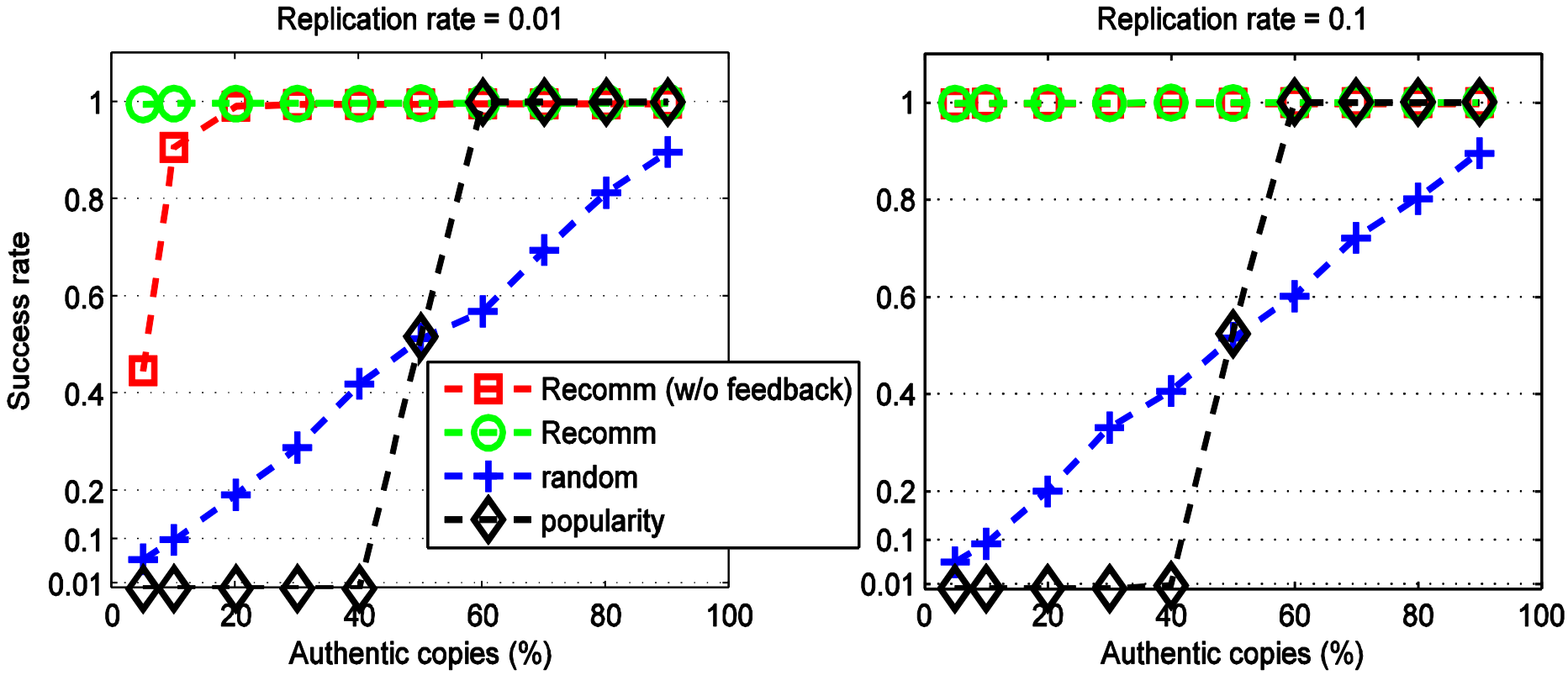
L = 12 (w/o overriding policy)

(a set of honest publishing nodes with 108 links are chosen)

Success rate vs. # of good publishers



Comparison with Heuristic Approaches in 10,000 P2P Networks



Conclusions

- Proposed a recommender to defend against pollution in P2P networks
- Exploited trust links on OSNs to manage user opinions on previously downloaded files and to provide recommendation in file downloading
- Showed effectiveness of the recommender