UNIVERSITY OF CALIFORNIA
UCRIVERSIDE

Decentralized Trust Management for Ad-Hoc Peer-to-Peer Networks

Thomas Repantis *Vana Kalogeraki*

Department of Computer Science & Engineering
University of California, Riverside

{trep,vana}@cs.ucr.edu

http://www.cs.ucr.edu/~{trep,vana}

Ad-Hoc Peer-to-Peer Networks

- Personal mobile devices can form ad-hoc networks to autonomously share data and services
  - ***Work-related projects***
  - ***Multi-player games***
  - ***Social networks***
  - ***Auctions***
- Nodes are both clients and servers
- No central coordinator

Advantages of Peer-to-Peer

- Scalability: No central coordinator
- Reliability: No single point of failure
- Self-organization: Autonomous decisions to adapt to different loads
- Resource aggregation: Take advantage of existing resources
- Successfully deployed for:
  - Distributed Computing (e.g. Seti @, Folding @)
  - File Sharing (e.g. Gnutella, DHTs)
  - Online Gaming (e.g. Playstation)
  - Spam Detection (e.g. SpamNet)

Our Research Question

- How to enable a peer to decide whether to trust another peer in the absence of a central trust managing authority
- A puts a level of trust into B means that A estimates the probability of B acting in a way that will allow A to achieve a desired level of satisfaction
- A can estimate the level of trust to put into B based on B's reputation, built from B's previous interactions
- Challenges:
  - ***Information about peer interactions is spread across the network***
  - ***Malicious peers might tamper with reputation information while stored or transmitted***

Reputation-Based Trust Management Middleware Requirements

- Enable peers to identify trustworthy peers for the particular resource and level of trust they require
- Light-weight, so that the protocol overhead is not hindering peers' interaction
- Resistant to reputation tampering
- Resistant to collusions

Our Approach

- Decentralized trust management middleware for unstructured, ad-hoc, peer-to-peer networks, based on reputation
- Storing the reputation information of a peer in a group of peers not easily identifiable, i.e., its neighbors
- Reputation piggy-backed on a peer's replies
- Taking advantage of the lack of network structure to resist collusions and blackmailing

Roadmap

1. Motivation and Background
2. System Model
3. Operation
4. Attacks
5. Algorithms
6. Experimental Evaluation
7. Related Work
8. Conclusions and Future Work

System Model

- Peers identified by public/private key pairs
- Provide objects (data or services)
- Form unstructured, self-organizing network
- Peer offering an object receives a rating r
- Reputation R is the sum of ratings
- Consumer trusts provider if its reputation is higher than the minimum trust level it requires for this particular type of object

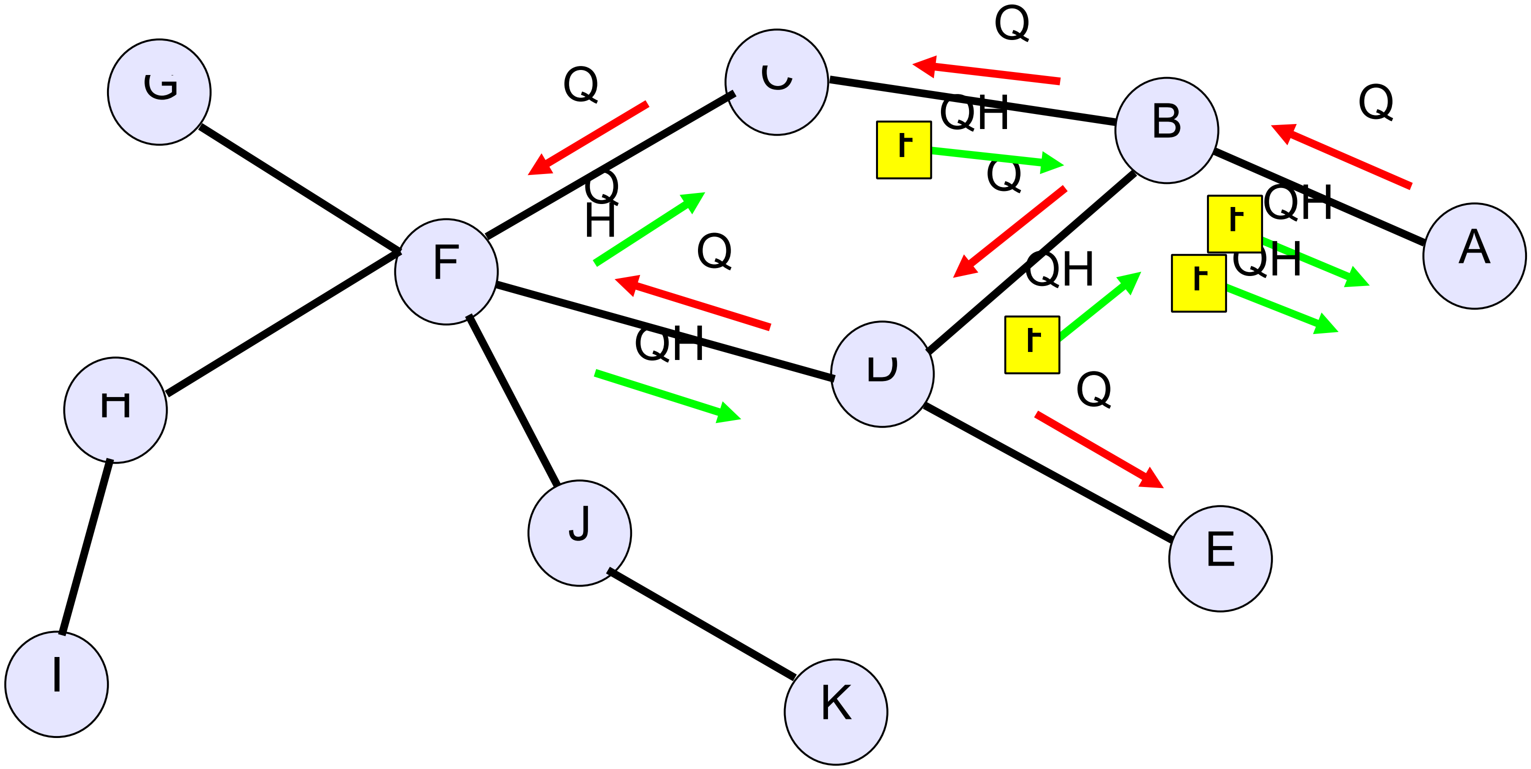
Object Discovery

- Peers search for objects by sending queries to their immediate neighbors
- Queries are propagated until their TTL expires
- Matches generate query-hits
- Every query is identified by a transaction globally unique identifier, TID
- TID is a random number together with the public key of the peer that produced the query
- TID is the same for the query, all query-hits, and all ratings produced as a result of the query
- By caching TIDs, query-hits follow the reverse path of the corresponding queries

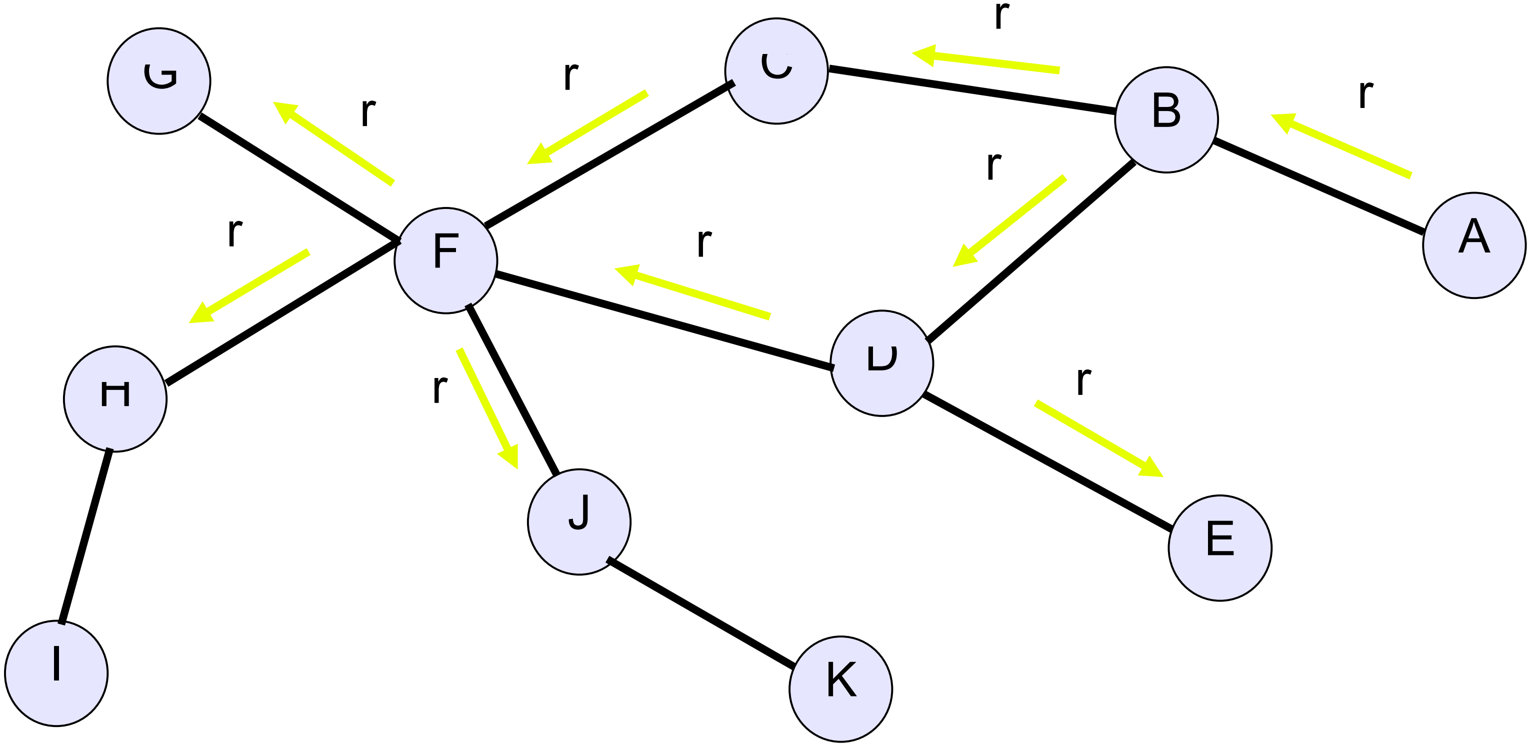
Reputation Propagation

- Every immediate neighbor of a peer, through which a query-hit of the peer travels, is responsible for piggy-backing the reputation of the peer to the query-hit
- All immediate neighbors are responsible for maintaining and piggy-backing its reputation
- The reputation reported for a peer is associated with a confidence value, determined by the number of neighbors reporting it
- After an interaction the consumer sends a signed rating to all producer's neighbors
- TTL of rating is larger than TTL of query by 1
- Rating is verified using the public key contained in query's TID

Query and Query-Hit



Rating

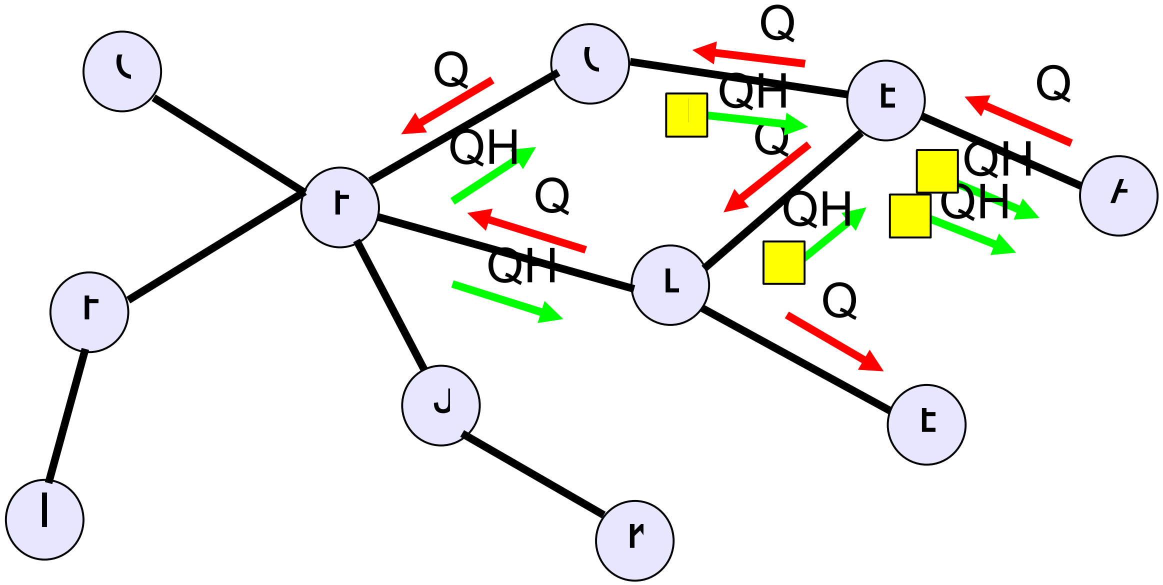


Roadmap

1. Motivation and Background
2. System Model
3. Operation
4. Attacks
5. Algorithms
6. Experimental Evaluation
7. Related Work
8. Conclusions and Future Work

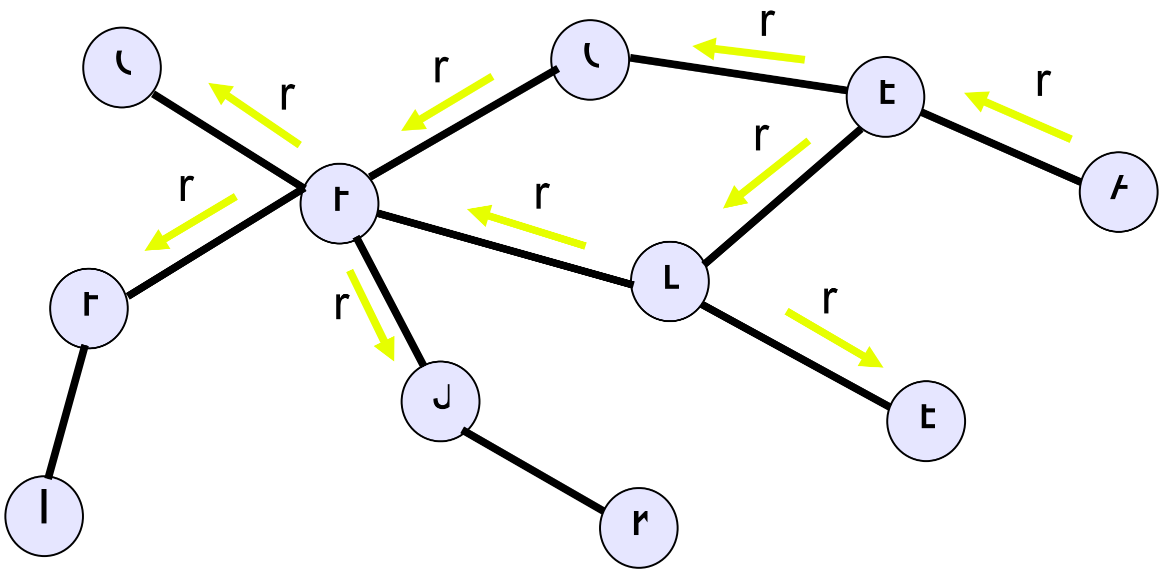
Against Tampering

- Attack: Alter neighbor's reputation
- Countermeasure: Since multiple peers might report a peer's reputation, tampering can be detected. A makes sure reputation of F reported by C and D is the same



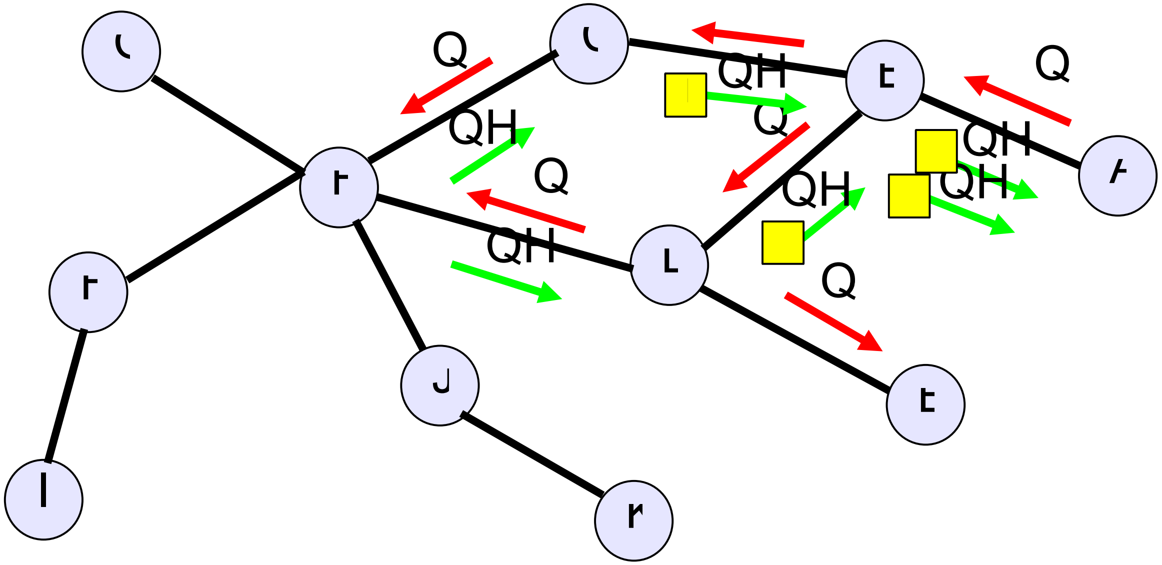
Against Tampering

- Attack: Alter own reputation
- Countermeasure: A peer does not store its own reputation
- Attack: Alter rating during transmission
- Countermeasure: Ratings signed by their creator



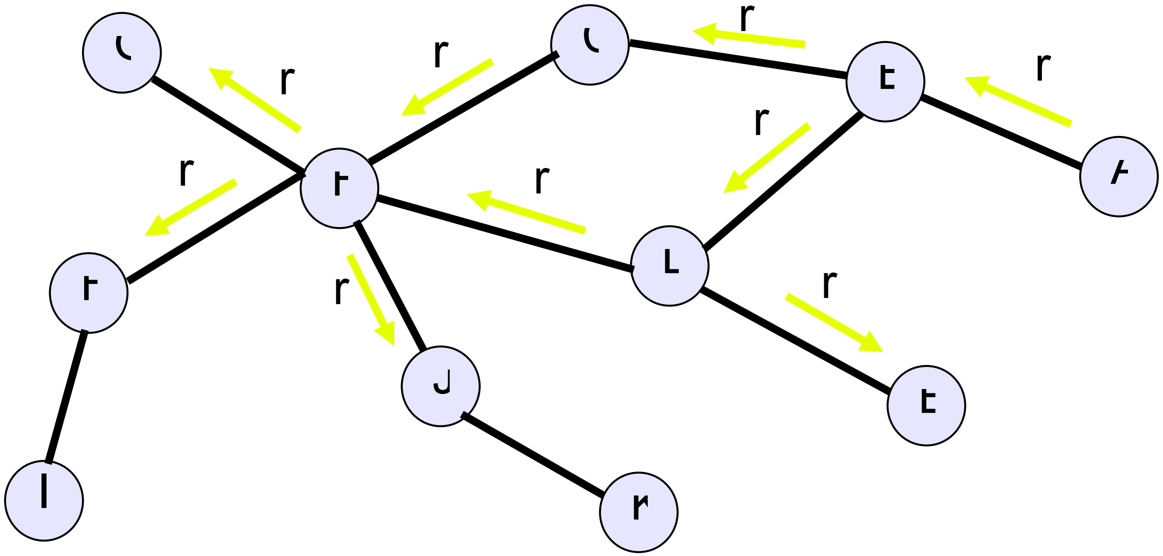
Against Blackmailing

- Attack: Peer blackmailing a neighbor to boost its reputation
- Countermeasure: Peers store their neighbors' reputation and their neighbors store theirs. Single neighbor reporting bogus reputation runs the risk of identification



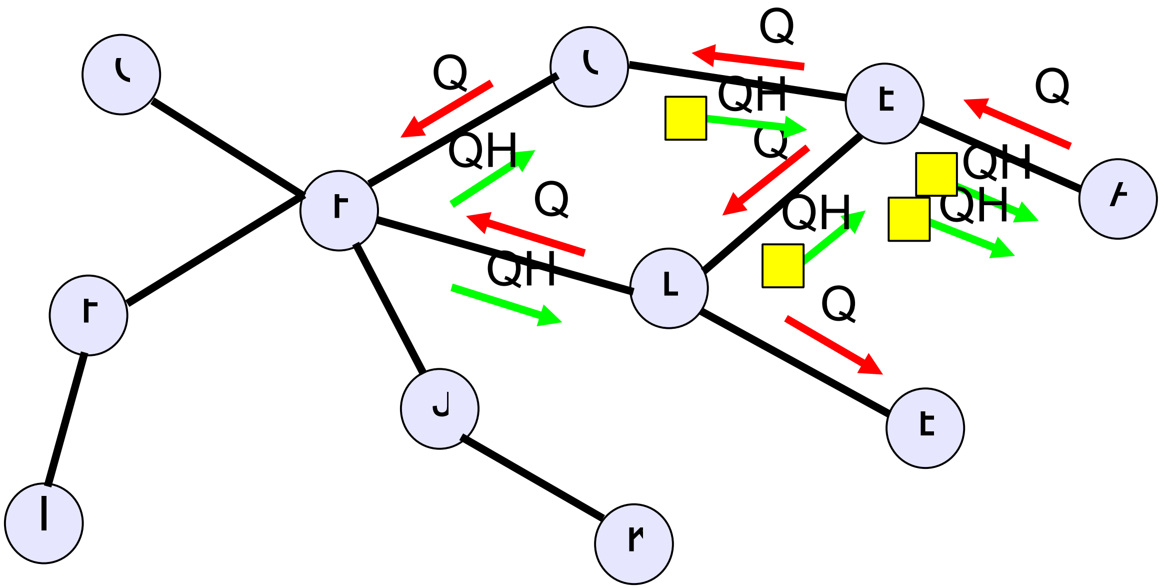
Against Multiple Ratings

- Attack: Submitting multiple positive or negative ratings
- Countermeasure: No effect, because no corresponding TID stored at the neighbors by a previous query-hit



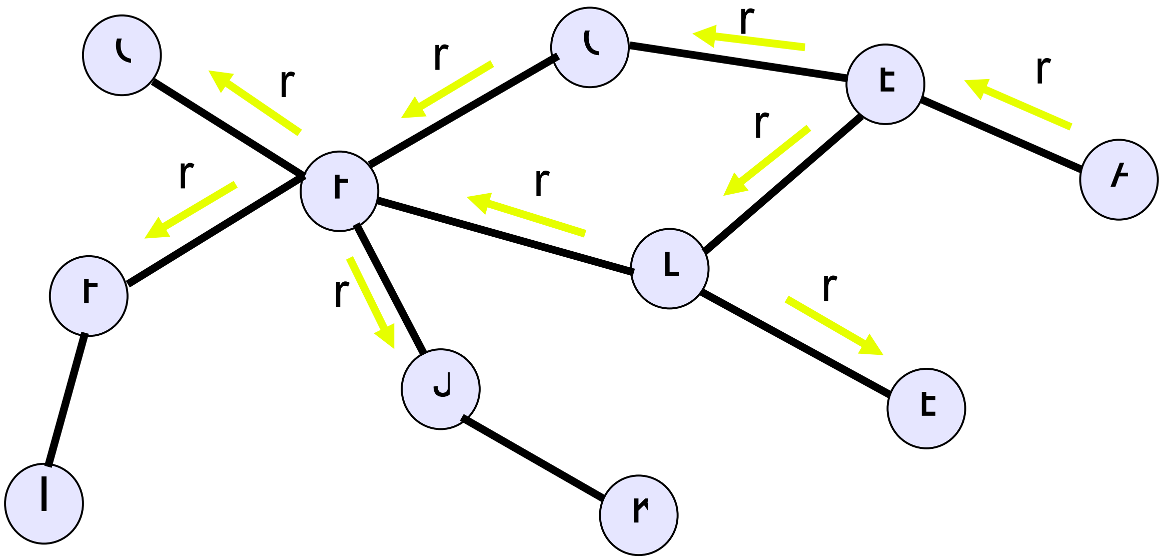
Against Collusions

- Attack: Two neighbors boosting each other's reputation
- Countermeasure: Would have to cooperate with all their neighbors and they consequently with all their neighbors etc.



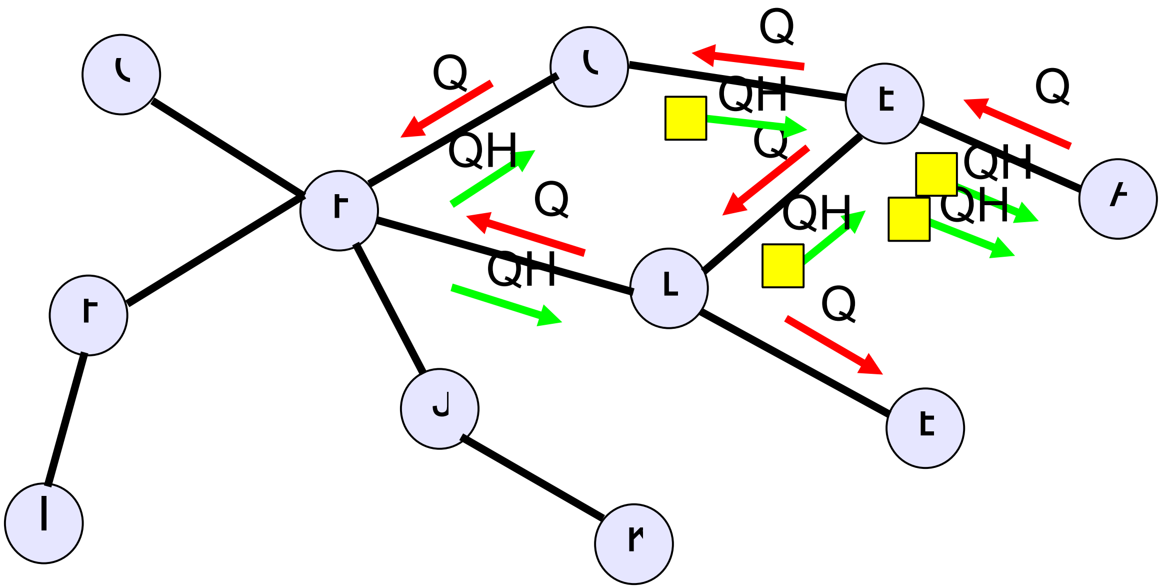
Against Collusions

- Attack: Peer bribing some of its neighbors to boost its reputation and only propagating query-hits through them
- Countermeasure: Detected by the rest of the neighbors when receiving unexpired ratings for their neighbor, with TIDs of query-hits they had not propagated



Against Collusions

- Attack: Peer bribing all of its neighbors to boost its reputation
- Countermeasure: A high confidence value requires a high number of bribed neighbors



System Algorithms

- Selection Algorithm:
- Per object trust and confidence levels
  - $R_i \geq L_j$
  - $C_i \geq K_j$
- Rating Algorithm:
  - Binary rating scheme, -1 dissatisfied, +1 satisfied
  - Enable objective interpretation and automatic assignment
- Initialization Algorithm:
- Protocols against sybil attacks can be integrated in our middleware to prevent identity changes

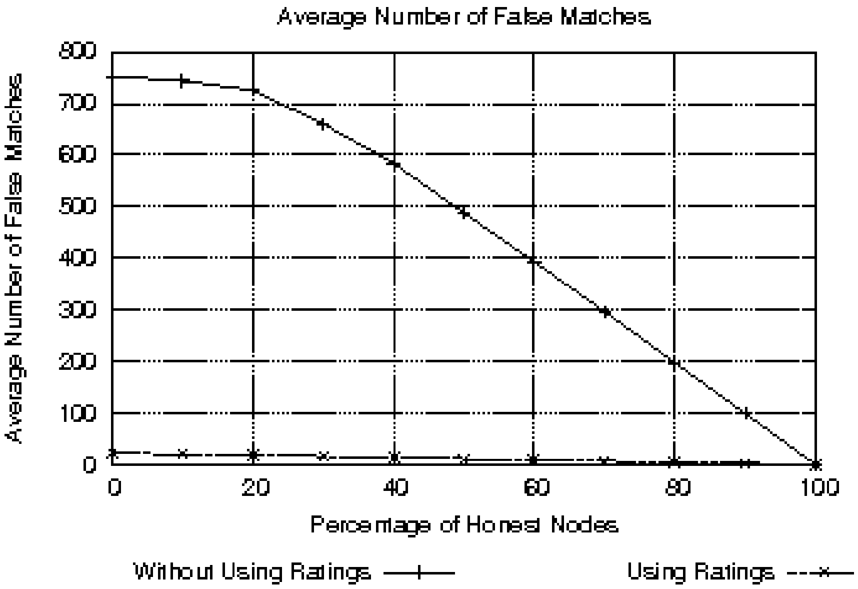
Roadmap

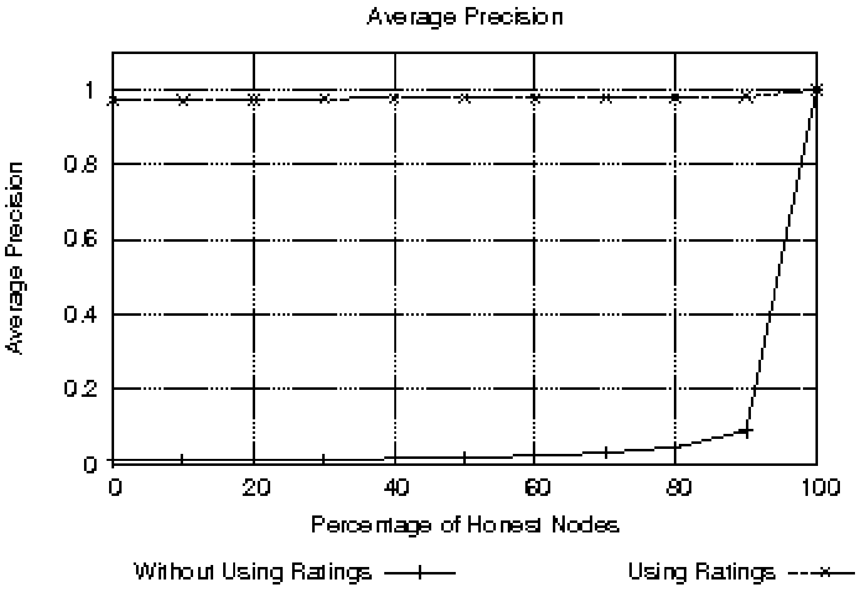
1. Motivation and Background
2. System Model
3. Operation
4. Attacks
5. Algorithms
6. Experimental Evaluation
7. Related Work
8. Conclusions and Future Work

Experimental Evaluation

- Simulated Gnutella unstructured, peer-to-peer networks of thousands of peers using NeuroGrid simulator
- 3000 types of objects, 30 objects per peer
- 100 random searches per experiment and average results from 5 measurements
- Malicious peers claim they have every object they are asked for but they can only cheat undetected once

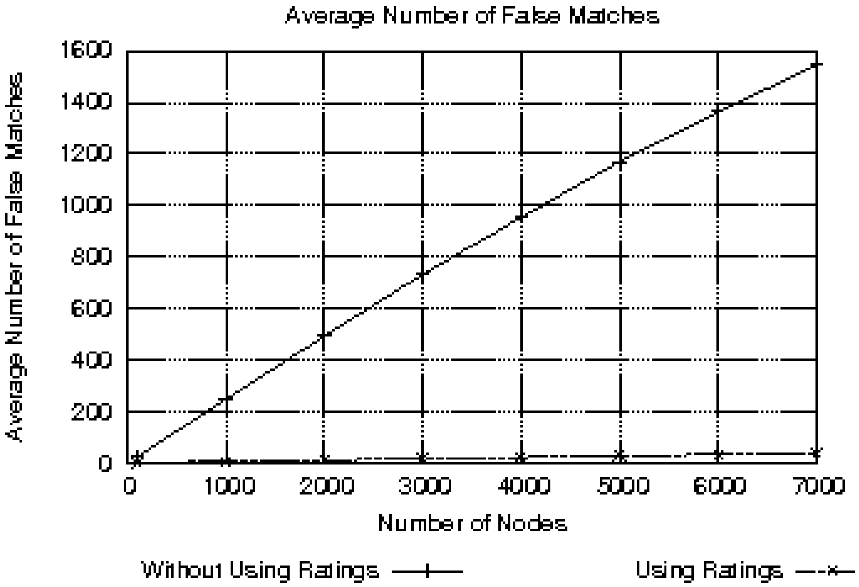
Variable Percentage of Honest Peers

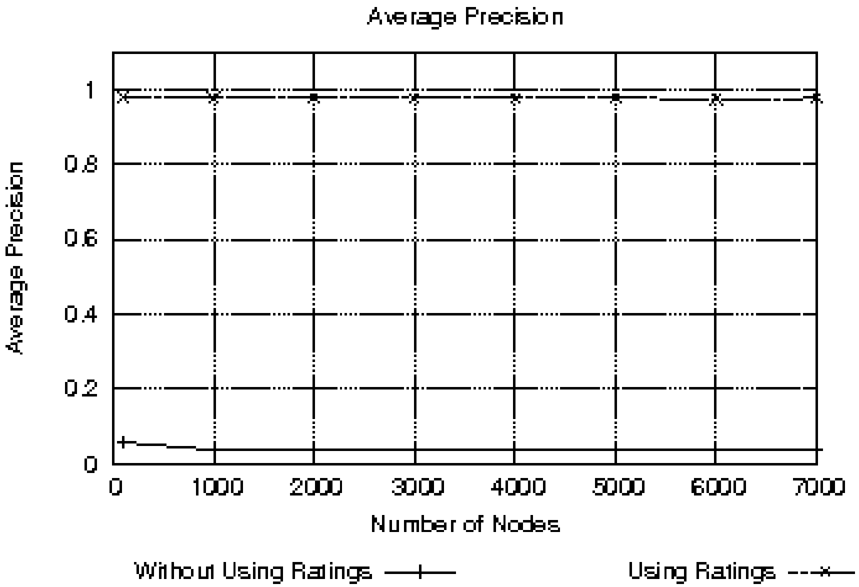




- If 1 out of 10 peers is dishonest, 9 out of 10 query-hits are bogus

Variable Number of Peers





- Dishonest peers can flood even networks of thousands of peers

Related Work

- Peers polling for opinion of others: *P2PRep*
- Reputation certificates signed by raters: *RcertPX*
- Reputation stored in anonymous random peers: *TrustMe*
- Reputation replicated in a group of peers: *EigenTrust*
- Voting on the reputation of objects instead of peers: *Credence*
- Identify ratings not corresponding to actual transactions: *TrustGuard*

Conclusions and Future Work

- Decentralized trust management middleware for ad-hoc, peer-to-peer networks, based on reputation
- Takes advantage of unstructured topology to make malicious behavior risky
- Peers are equal and self-organizing
- Fully distributed, non-intrusive protocol
- Future work: Investigate the effects of mobility, elaborate on peer selection and rating algorithms

Thank You!

http://www.cs.ucr.edu/~trep/

Decentralized Trust Management for Ad-Hoc Peer-to-Peer Networks

Thomas Repantis *Vana Kalogeraki*

Department of Computer Science & Engineering
University of California, Riverside

{trep,vana}@cs.ucr.edu

http://www.cs.ucr.edu/~{trep,vana}