Accountability and Resource Management

A discussion of issues
for peer-to-peer systems

Roger Dingledine
Reputation Technologies
arma@reputation.com

Michael J. Freedman
MIT LCS
mfreed@mit.edu

The Free Haven Project
freehaven.net

The Resource Management Problem

- Goal: maximize a peer's *utility* to the overall system while minimizing its potential *threat*.

- Threat: peers eat resources

Managing scarce resources…

- Freenet: unpopular data is dropped; popular data is cached near the requester
- Gnutella: data is stored only on the publisher’s own computer
- Publius: currently limits submissions to 100K

Accountability

- Approach to resource management
- Resources more efficient and protected

Why is P2P accountability hard?

- Tragedy of the commons
- P2P discourages permanent public identification
- Hard to assess peer’s history or predict future performance
- Legal contracts are outdated and impractical

Introducing accountability…

- Mojo Nation: micropayments are used for all peer-to-peer exchanges

- Free Haven: reputation system – publishers must provide reliable space of their own

- Mixmaster: statistics pages track uptime

Discussion outline

- Accountability problem
- Current systems
- Models of P2P systems
- Resource management techniques
  - Electronic payments
  - Reputation systems
- Conclusions

Problems to tackle

- Intentional attacks (adversaries) and simple overuse (freeloaders)

- User attacks
  - Communication DoS (query flooding)
  - Storage flooding
  - Computational overload

Problems to tackle

- “Server” attacks – low-quality service
  - Dropping data
  - Providing corrupted data
  - Ignoring requests
  - Going down when needed
  - Adversarial collusion

…not following system protocol !

Problems in current P2P systems

- Freenet
  - Bandwidth overuse (query flooding)
  - Cache flushing (data flooding)
- Gnutella
  - Vulnerable to query flooding
  - Freeloading
- Publius
  - Public server identities:
    directed attack on bandwidth, storage space

Problems in current P2P systems

- Mojo Nation
  - How to set prices?
  - Performance tracking, not reputation
- Free Haven
  - Very vulnerable to query flooding
  - Protected against data flooding
    (reputation system is complex and untested)
- Mixmaster
  - No verifiability
  - Uptime is not reliability

Two accountability solutions

- Restrict access to resources
  - Digital payment mechanisms
- Select favored users
  - Reputation schemes

P2P models

1. Static, identified operators
   - Examples: Mixmaster remailer, Publius
   - Limited users: legal mechanisms possible
   - Reputation and payment schemes

2. Dynamic, identified operators
   - Examples: Gnutella, Freenet, Mojo Nation
   - Reputation and payment schemes

P2P models

3. Dynamic, pseudonymous operators
   - Example: Free Haven
   - Reputation and payment schemes
     - Decisions may be based on prior behavior

4. Dynamic, anonymous operators
   - Payment schemes only
     - All information is ephemeral
     - Decisions based only on current transaction

Goal of payment schemes

- Manage scarcity of resources
  - Charge for access
- Prevent intentional attacks
- Restrict freeloading
- Result: optimize for “social efficiency”
  - Users contribute to overall system robustness

Payment schemes: models

- Proofs-of-Work (POWs)
  - Examples: hash cash, Client Puzzles
- Fungible non-anonymous payments
  - "Credit cards"
  - Examples: MicroMint, PayWord, Millicent, Mondex
- Fungible anonymous payments
  - "Cash"
  - Examples: Chaum's eCash, Brands' digital cash

Payment schemes: distributed use

- How to stop double spending?

- Centralization: central “bank” servers
  - Support balance transfer
  - Fungible payments

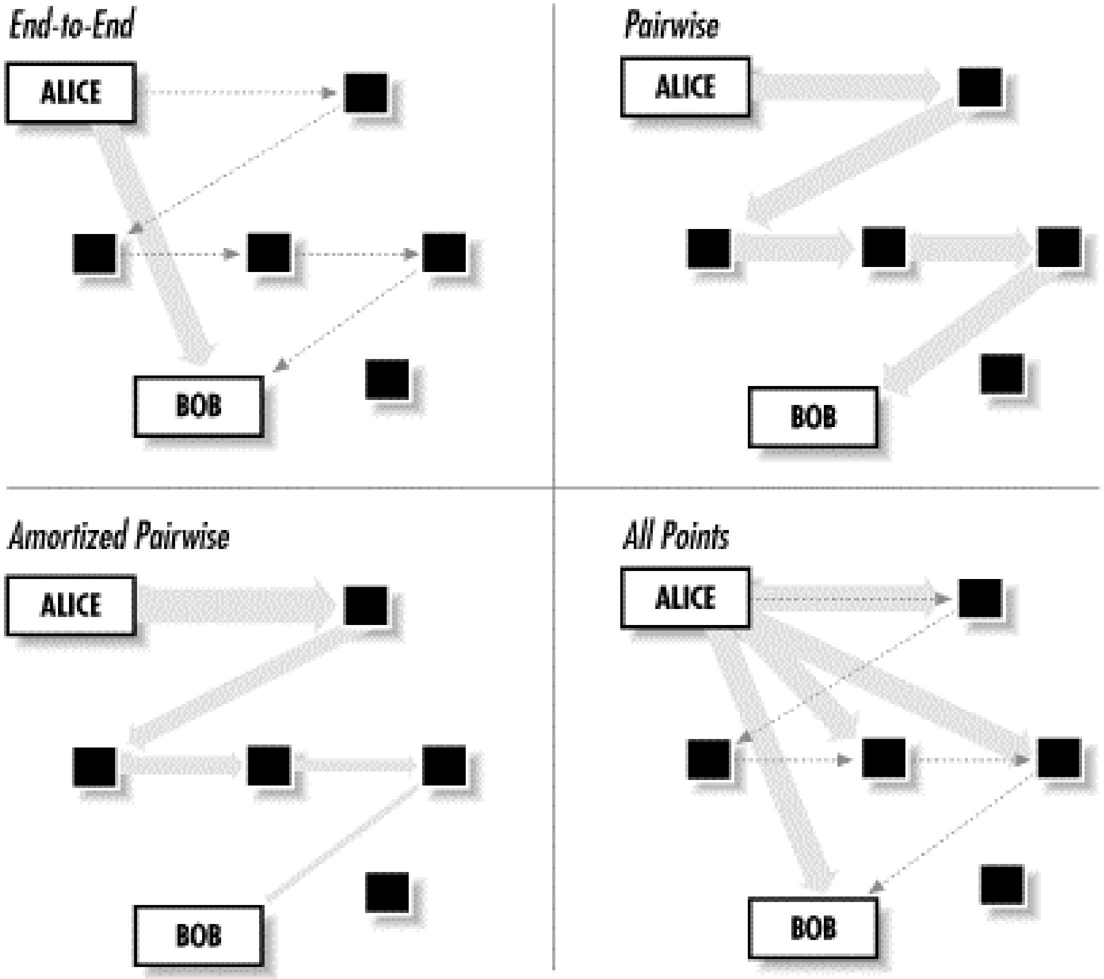
- Decentralization: recipient-specific payments
  - POWs encode recipient in solutions
  - Peers issue “own” currency

Congestion management

- Renewable resource allocation
  - – Determine need dynamically
  - – Areas: bandwidth, computation, caching
  - – *Solution? Only charge when congested*

- Cumulative resource allocation
  - – Once allocated, not easily recoverable
  - – Area: persistent storage
  - – *Solution? Always charge*

Payment models



Example: Anon communication

- Java Anon Proxy
  - – Stop message flooding by recipient-specific tickets
  - – All-pairs: O(*mn*) tickets, *m* edges, *n* core nodes
  - – Proactively manages resources

- Reactive bandwidth throttling
  - – Recipient-specific proofs-of-work
  - – Pairwise for real-time connection-based networks

Example: Pseudonymous storage

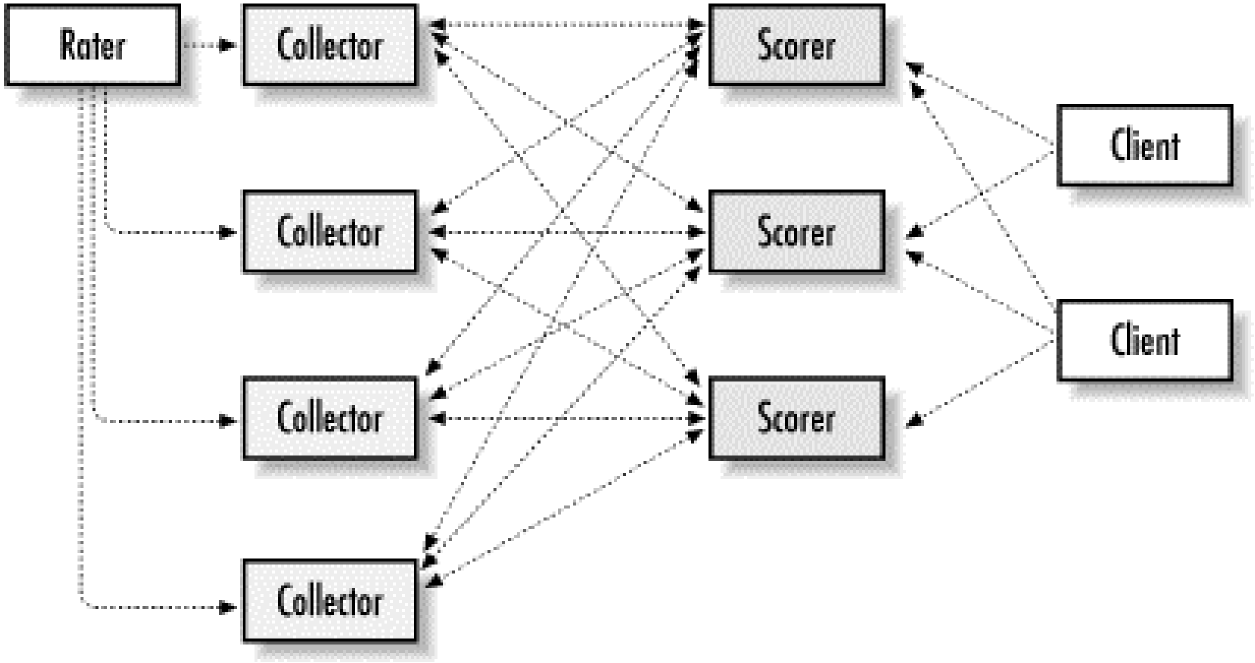
- “Eternity Service”
  - All-or-nothing!
    - Pay servers fungible lump-sum
  - Reward partial work?
    - Small payments per time-slice
  - How ensure servers respond to requests?

$\Rightarrow$ reputation systems

Reputation systems

- Track performance to predict future behavior
- Risk resources based on anticipated benefit (resource management approach)

Reputation systems



- Information provided by third parties

Example reputation systems

- PGP Web of Trust
  - Does not actually map key to *person*
  - Scalability? graph not dense enough
  - Certification to do what?
- Advogato
  - Uses maximum flow to calculate reputation
  - Three levels of certification: apprentice, journeyman, master
  - Resists pseudospoofing via trust bottlenecks

Example reputation systems

- eBay
  - – Collects feedback about transactions
  - – Small sales treated same as large
  - – Almost no negative feedback given!
- Google, Clever
  - – Many pages point to you ⇒ popular
  - – Popular pages point to you ⇒ credible
- Mojo Nation
  - – Remember how nodes treat you (performance, accuracy)
  - – Hard to tune prices?

Example reputation systems

- Mix-net reputations
  - Scorers track delivery failures, publish reputations
  - Need to tune parameters, e.g., how long nodes remember ratings
  - Higher reputation draws more traffic
- Free Haven
  - Need to notice servers that drop data early
  - Need mechanism to “punish” misbehaving servers
  - Nodes periodically broadcast reputation referrals
  - Credibility different from reputation

Some goals for reputation systems

- Local / personalized reputation
- Resist pseudospoofing
- Resist shilling, e.g., verify transactions
- Collect enough data to be useful
- Distinguish between reputation and credibility

Accountability slider

- Dynamically determine need and extent
- Digital payments
  - Adjust “amount” charged
- Reputation systems:
  - Adjust “trust” thresholds

Conclusion

- Peer-to-peer won't save you
- Accountability is not pixie dust
- Payment and reputation systems are efficient and flexible solutions
- Verifying behavior still necessary
- Convenience trumps accountability…

Further reading…

Peer-to-Peer:

Harnessing the Power of Disruptive Technologies

Chapter 16: Accountability

The Free Haven Project

freehaven.net