Reputation Management in P2P Systems

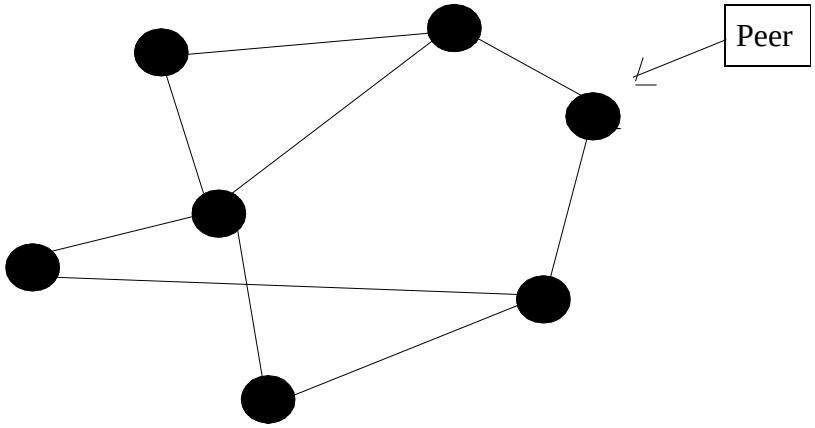
Pradipta Mitra
Nov 18, 2003

We will look at ...

- Overview of P2P Systems
- Problems in P2P Systems
- Reputation Management
- Limited Reputation Sharing
- Simulation Results
- Global Reputation Sharing

P2P Systems

- Individual Computers connected in a overlay network
- Objective: Resource(File) Sharing
- Popularity of P2P is increasing
- Examples: Gnutella, Chord, CAN, Freenet



A P2P System

P2P Systems(Continued)

- Autonomy
- Anonymity
- Decentralization
- Openness

These attractive features are the very root of its problems!

Problems in P2P Systems

- Free Riding!

"...almost 70% of Gnutella users share no files, and nearly 50% of all responses are returned by the top 1% of sharing hosts"
[Study by Abar and Huberman (2000)]

- Uploading at slow rates

A recent product, Bit Torrent claims to solve this problem.
"On average, the faster you upload to your peers, the faster you will be able to download."
[http://bt.etree.org/]

But Selfishness is not the big problem

The BIG PROBLEM is Maliciousness!

"... peers that volunteer to share files are not necessarily those who have desirable ones. ... [this]adds vulnerability to the system."

[Study by Abar and Huberman (2000)]

- There might be nodes in the system who don't want to access other people files, rather only are interested in propagating there own invalid files.

Maliciousness introduces two problems:

- Document Authenticity Problem: We shall not handle this!

- Detecting Malicious peers: Creation of a **Reputation System**.

Design Considerations

- Self-policing
- Anonymity Maintaining
- Minimal Overhead
- Robust to Collectives

The System Model

- When a node queries the system for a file, it collects all replies in a *response set.*

- The node repeatedly selects responses from the set, fetches the file and verifies it until an authentic copy is found.

- A system without reputation management will select peers randomly.

- The goal is to create reputation for nodes so that the number of inauthentic files downloaded is minimized.

The Threat Model

1. *Individual Malicious peers* who are somewhat stupid: They always provide an inauthentic file when selected as a download source.

2. *Malicious Collectives*: Similar as (1) but provides high reputation values for each other and low for all others.

3. *Malicious Collectives with Camouflage*: Provides inauthentic files in *f%* of cases that it is selected.

4. *Malicious Spies*: Themselves provide good files but try to increase the reputation of other malicious peers.

5.Good nodes unknowingly providing bad files.

Three Approaches

- Global Reputation System: Ask Everyone.

- Limited Reputation Sharing System
  - Local Reputation System: Self-Help is the Best Help.

  - Voting Reputation System: Ask for opinions of some other peers.

Local Reputation Systems

Reasons for Considering Local Reputation: Calculating a consistent global reputation is costly.

- Each node maintains a statistics on how many files it has verified from each peer and how many of those were authentic.

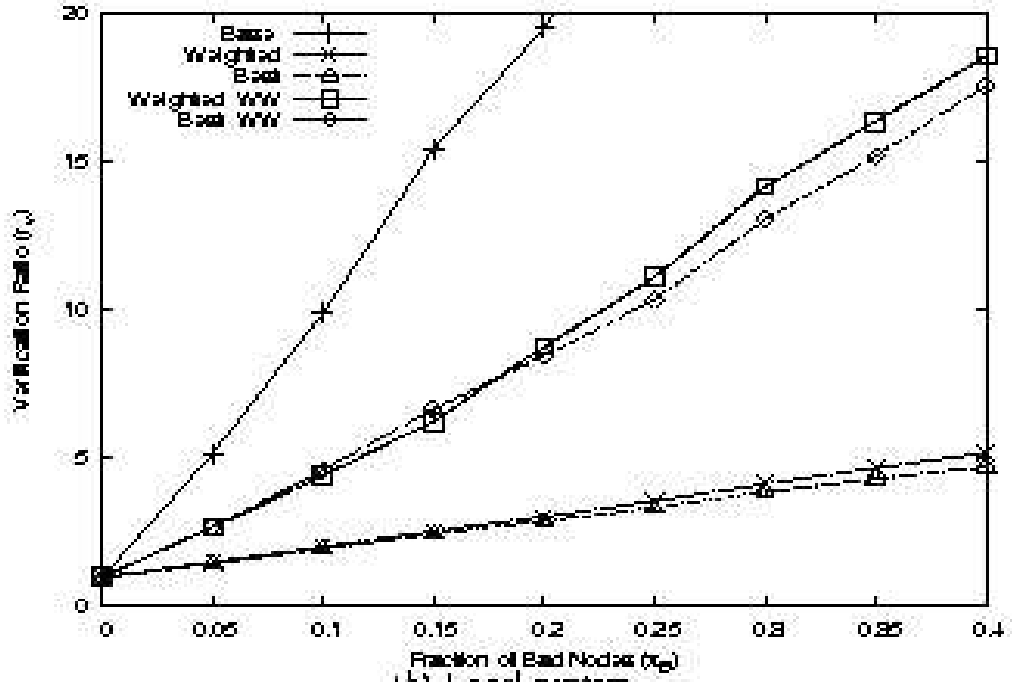
- Each peer's reputation rating is calculated as the fraction of verified files which were authentic. This results in a rating ranging from 0 to 1.

- Doesn't distinguish between bad nodes and nodes with which it hasn't interacted.

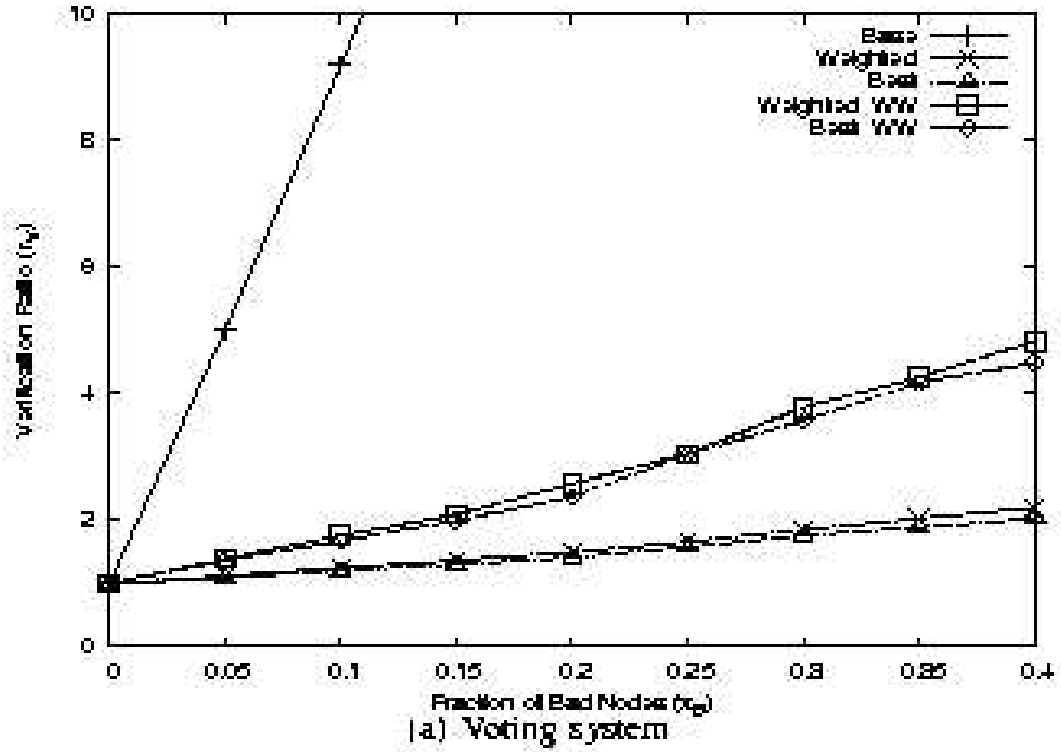
- Two methods of choosing from a response set:
  - Select-best Selection Procedure: Has the problem of overloading the best nodes.

  - Weighted Selection procedure: Select a peer probabilistically. The probability of choosing a peer will be proportional to its reputation.

Simulations show that the weighted method distributes load better without degrading efficiency much.



(b) Local system



(a) Voting system

Friend-Cache in Local Reputation Systems

- In Select-best method a node maintains an ordered list of most reputable nodes it knows.
- This list is known as the friend-cache and has a maximum size *FC*
- The *Friend-First* technique uses the

Friend-Cache by querying the friends first before querying the network in the usual fashion. Reduces the message traffic by upto 85%.

Voting Reputation Systems

- Extends local reputation systems by considering opinions of other peers in the selection stage.
- When a node q has received a set of responses to a query, it contacts a set of nodes Q, for their opinions of the responders.
- The final rating for each responder p_r is calculated by:

$$p_r = (1-w_Q)R(q,r) + w_Q\left(\sum_{v\in Q}R(q,v)R(v,r)\right)/\sum_{v\in Q}R(q,v)$$

R(a,b): node a's opinion of b.

- For each “voter”, the opninion of the voter is weighted by the reputation of

that voter.

- The final rating is a combination of local experience and peer responses. These two values are combined using the quorumweight w_Q.

- When $w_Q = 0$, the voting reputation system becomes the local reputation system.

- The select-best procedure and the weighted procedure apply to voting reputation systems as well.

How to Select Voters?

- Neighbor-voting: Asking opninions from ones neighbors in the overlay network. Number and Identity of voters remain *relatively* constant.

- Friend-voting: Ask peers from whom one has fetched files and who have proven to be reputable. The friend-cache can be used.

Before discussing simulation results, there is one big issue to resolve.

What else but ....

IDENTITY!

Anonymity allows for various attacks on a P2P system:

- Sybil attack: Automatically creating indentities. Solvable using "*captcha*".
- Whitewashing: Malicious peers change their identities when their reputation goes down.

The second problem can be tackled by two different approaches:

- Centralized Login Server: Real world identity tied to one unique system ID.

- Makes Changing Identities difficult (if not impossible)
- However, contrary to the spirit of P2P.

- Assigning new nodes low reputation ranking: This eliminates the incentives of changing identities. However, again somewhat contrary to the spirit of P2P.

Unfortunately, if we allow users to change identity...

"*No strategy ... can do substantially better than punishing all newcomers.*" [Friedman and Resnick, 2000]

- No definitive solution found so far. Reserachers have proposed Initial reputation $p_0 = 0$.

Simulation Results for Limited Reputation Sharing

Metrics

- Verification Ratio,
  $r_v = (\sum_i V_i)/q_{succ}$

  - $\sum_i V_i$ *is number of file verifications done over all nodes*
  - q_{succ} *is the number of successful queries.*

- Relative Message Traffic

  MT_{rel} = *Nfriend/Nflood*

  Nfriend: No of FF Messages

  Nflood:No of Flooding Messages

Parameters

---

n = 1000

d_{max} = 50

d_{avg} = 3

Connectivity follows Power-law distribution

Number of copies of each file is also a Zipf distribution

For whitewashing $p_0 = 0$

Otherwise $p_0 = 0.3$

(Default) fraction of malicious nodes, $\pi_B = 0.3$

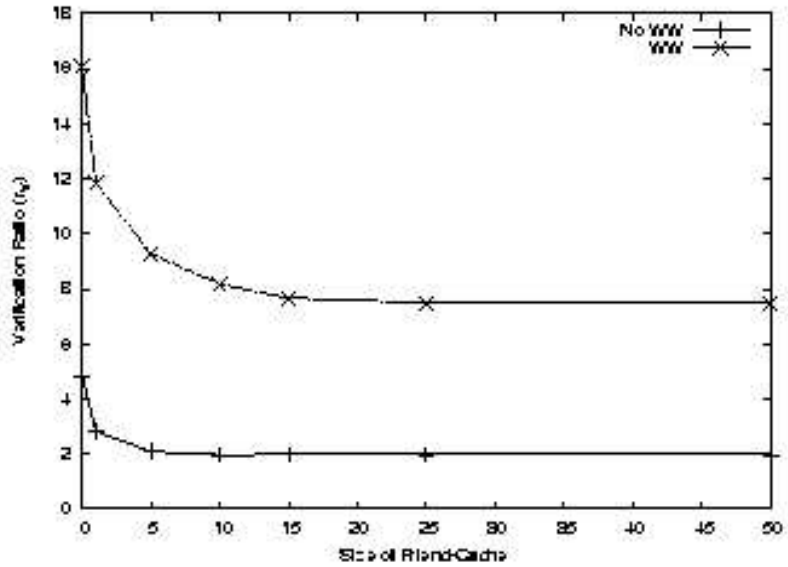


Fig. 2. Efficiency of the voting reputation system with respect to Friend-Cache size ($\overline{FC}$).

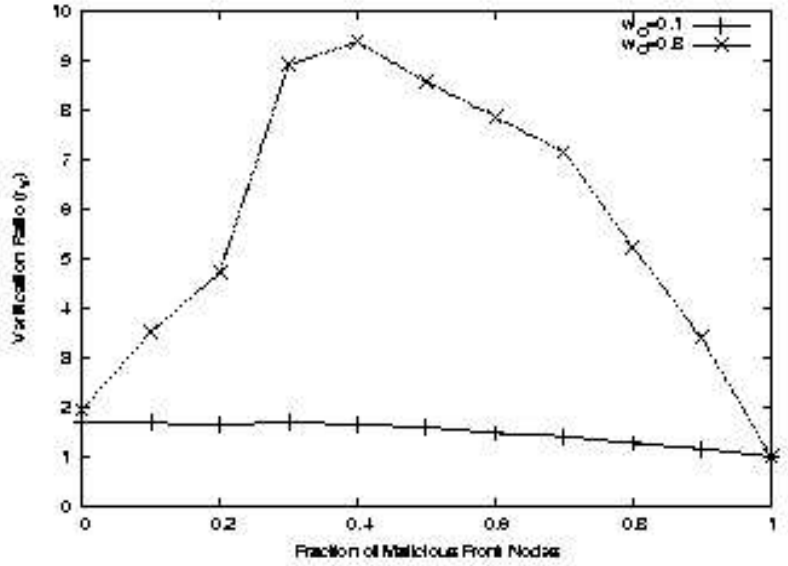


Fig. 3. Effects of front nodes on efficiency.

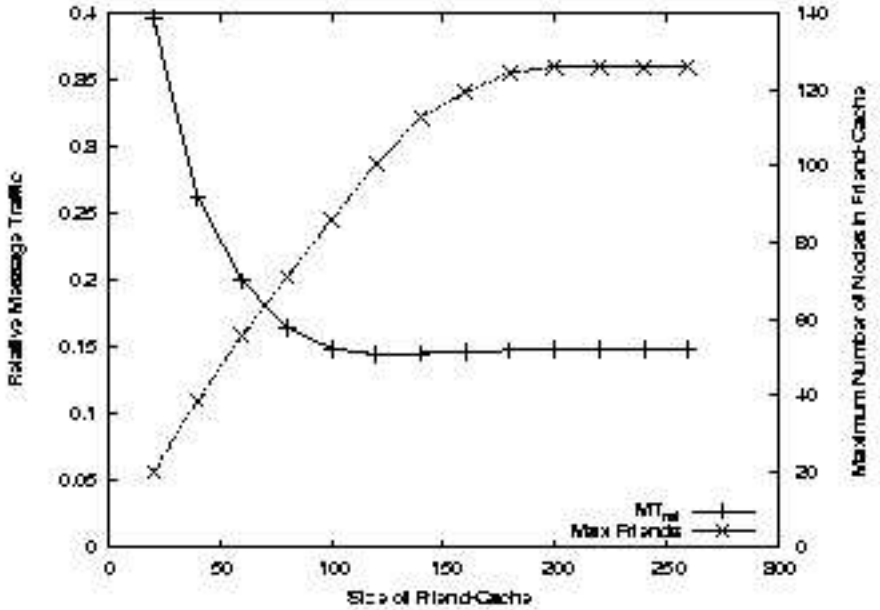


Fig. 6. Relative message traffic of Friends-First and maximum Friend-Cache utilization as a function of the size of the cache.

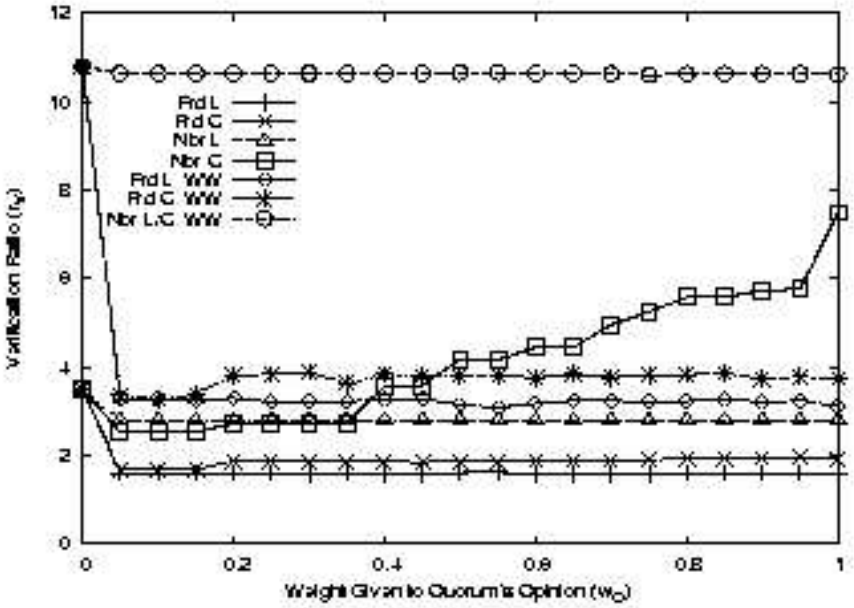


Fig. 1. Efficiency of the voting reputation system with respect to varying quorumweight (w_Q). Lower r_1 is better. 1 is optimal.

Global Reputation Systems

- Can be seen as an extension of the Voting Reputation System (where everyone votes)
- Clearly would have better performance, but large overhead

The Eigentrust Algorithm

c_{ij}=*normalized local trust value*

$t_{ik}=\sum_j c_{ij}c_{jk}$

$t_i=C^T c_i$

for some n, $(C^T)^n c_i$ will converge for all i.

The Eigentrust Algorithm(Contd.)

- If we had centralized server, it could

calculate the global trust values for us.

- Instead, we use distributed hash tables.
- Depends on the sparsity of the matrix for fast convergence.
- This makes Eigentrust usable only with complex P2P systems such as CAN and Chord.
- Still requires much more communication overhead than limited reputation systems.
- However, gives good performance where $\pi_B = 0.7$!

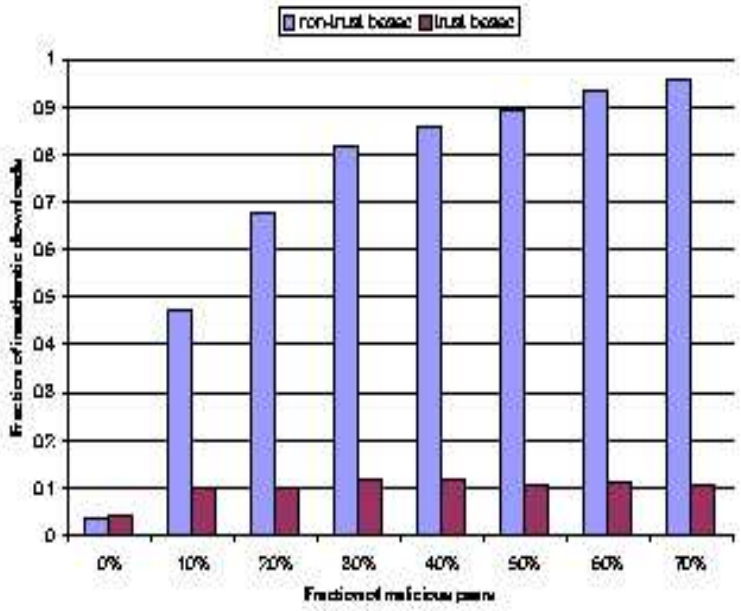


Figure 6: Trust-based reduction of inauthentic downloads in a network where a fraction of peers forms a malicious collective and always uploads authentic files. Forming a malicious collective does not boost the trust values of malicious peers significantly, they are still virtually banned from uploading inauthentic files, similar to Figure 5.

Challenges and Conclusion

- Handling Peers who are selfish in that they always choose select-best procedure.
- Devising global reputation system that works without using DHTs.
- Good nodes may inadvertently share invalid files. Allowing nodes to assign a confidence value to the files may be a useful.
- Examining the effect of weighting individual transactions in some way.
- Examining how long it takes for new

nodes to build up a reputation in self managed ID systems.

- Having partial authenticity instead of binary authenticity.

- Study so far is quite intuitive, specially the limited reputation schemes.

- Work done so far depends mostly on simulation results to prove their claims. Providing a theoretical framework is a big challenge.

References:

1. Kamvar, Sepandar D., Schlosser, Mario T., Garcia-Molina, Hector.: The EigenTrust Algorithm for Reputation Management in P2P Networks
2. Marti, Sergio; Garcia-Molina, Hector: Limited Reputation Sharing in P2P Systems
3. Marti, Sergio; Garcia-Molina, Hector: Identity Crisis: Anonymity vs. Reputation in P2P Systems