Quantitative Trust Management: Evaluating Trust/Reputation Systems

Insup Lee
Sampath Kannan
Oleg Sokolsky
Andrew G. West

Problem Statement

- There are many reputation/trust management systems already in existence
  - Applied systems use results from “independent” sims
  - Theoretical approaches may have no results at all
- ***Need a simulator for comparative analysis***
- Traditional network simulators are inappropriate
  - The complexity of simulating DHT’s, network hops, etc. PLUS the trust management calculations would prove a huge computational burden

Our General Solution

- We built our own trust management simulator along the trace/simulator paradigm
  - Simple network structure permits static analysis and minimizes complexity
  - Traces are highly configurable with regards to malicious users, file distribution, and bandwidth
  - Provides a simple interface to foster the development/installation of new trust algorithms
  - Traces are written to disk so they be analyzed under several different algorithms

High-Level Implementation

- Simulator basically in the P2P style
  - Users initialized with behavior model and (some) files
  - Particular copies of a file are strictly valid/invalid, though multiple copies may exist.
  - The transaction cycle:
    - Transactions begin with a query, which is answered only by potential sources with available bandwidth
    - Final source selection determined using trust values
    - Following transaction, receiver library has a copy of the requested file, and receiver provides either positive or negative feedback concerning transaction
    - Trust is re-calculated

Modeling User Behavior

- A two-dimensional approach to behavior:
  - Cleanup (%): Upon reception of an invalid file, how likely is it that a user will remove that file from their library
  - Honesty (%): With what probability will a user provide honest feedback

User Type	Cleanup	Honesty	Source
Good	90%-100%	100%	BEST
Purely Malicious	0%-10%	0%	WORST
Feedback Malicious	90%-100%	0%	RAND
Malicious Provider	0-10%	100%	WORST
Disguised Malicious	50%-100%	50%-100%	RAND
Sybil	Sybil users participate in 1 transaction then create a new ‘account’.		WORST

Note: “Source” dictates how trust values are used to choose a file-sender

Algorithms Being Studied

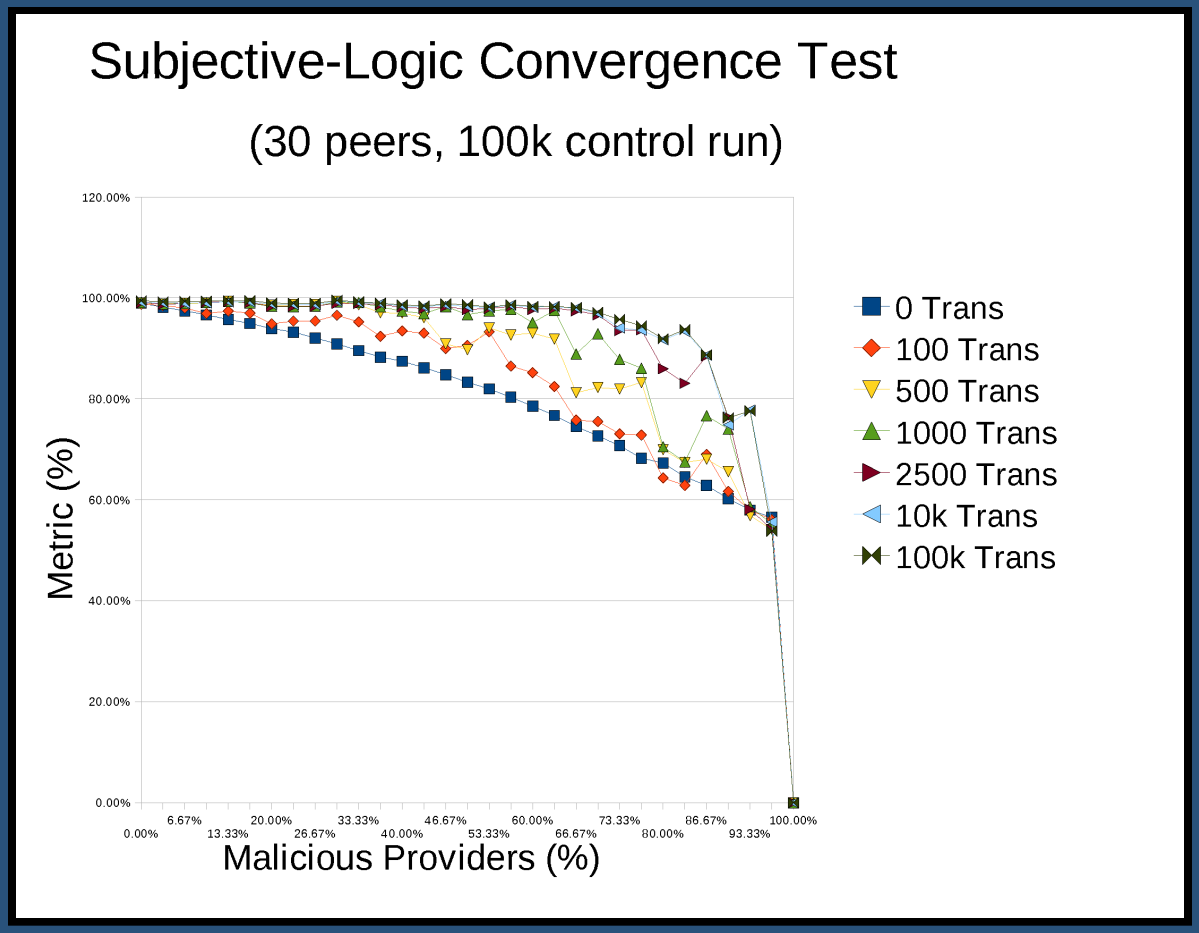
- *None*: The absence of TM, used for control runs
- *EigenTrust*: per Hector Garcia-Molina et. al.
  - Globally convergent trust via matrix multiplication of normalized values
  - Convergence quick due to certain matrix properties
- *Subjective Logic*: per Audun Jøsang et. al.
  - Triples of the form (belief, disbelief, uncertainty)
  - Transitive paths examined using ‘discount’ and ‘consensus’ logic operators
  - Trust values correlate with beta-PDF functions

Trust Algorithm Efficiency

- Problem: Matrix multiplication is complex, prohibiting experimentation with large networks (# of peers), and high transaction counts.
  - A 100k transaction, 30 peer SL simulation (≈ 10 min)
- Solution: Incremental trust computation
  - Difference vectors and vector-matrix multiply in place of matrix-matrix multiply, where possible
  - Consistent user behaviors create stable matrixes, thus trust need not be re-computed after *every* transaction
  - Current question: How often to re-compute?

How Often to Compute Trust?

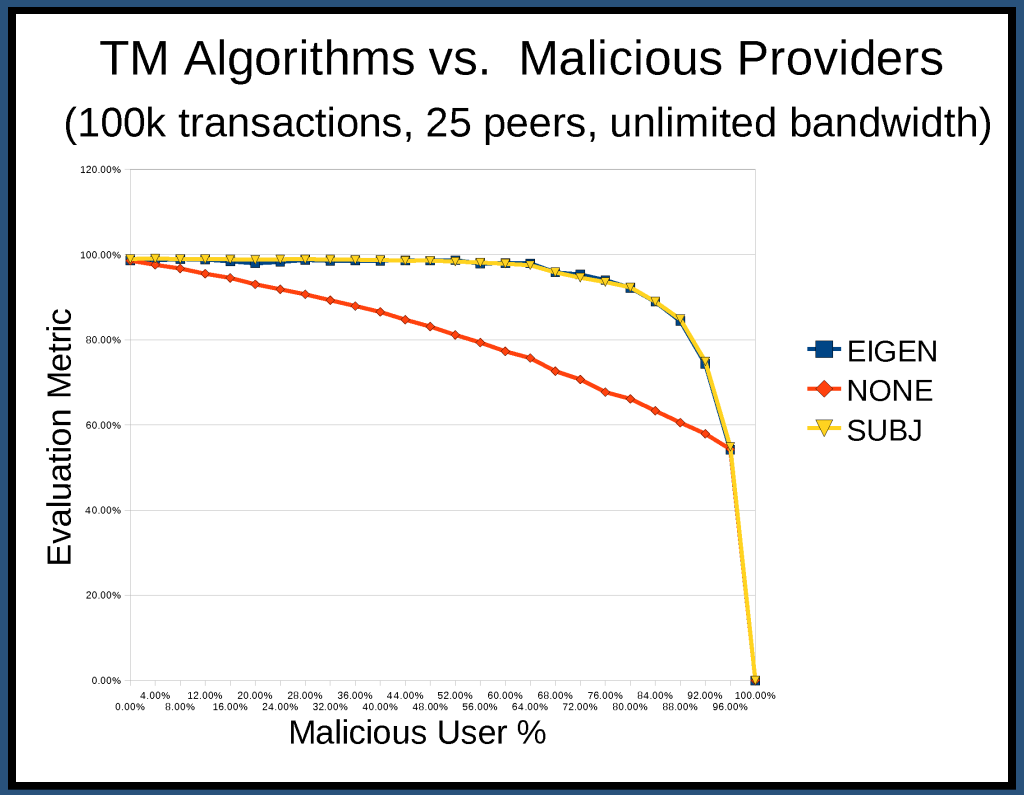
- Experiments show quick convergence and stability
- Only ≈3 transactions between each peer-peer relation is enough to “solve” matrix
- Implementing an adaptive approach to minimize computation under all runs, while maintaining correctness



Objective Function & Results

- Metric:

$$\frac{\text{(\# trans. with “good” recipients, resulting in trade of valid file)}}{\text{(\# trans. attempted by “good” users)}}$$

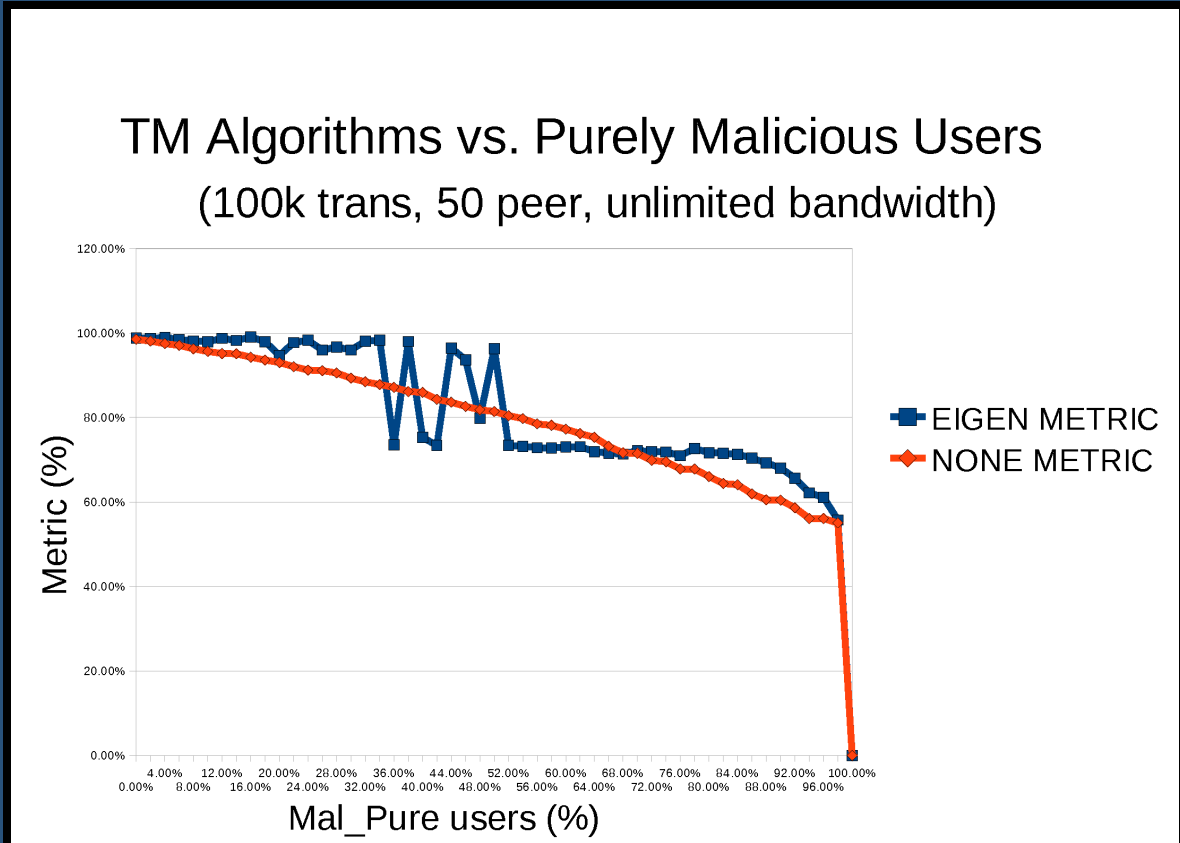


When everyone is honest about their behavior, as at left, it is trivial for TM-algorithms to show superiority

A More Complex Example

More interesting is when users are bad, lie about their behavior, and have other bad peers lie on their behalf (as at right).

EigenTrust in particular demonstrates some very interesting properties under varying network configurations (a topic currently under study).



General Conclusions

- Concerning simulator construction:
  - There is a huge space of network parameters; first-order behaviors need identified for inclusion
  - Efficiency is an issue in simulating larger networks
    - And many fixes could apply not just to simulation, but can be an improvement to the algorithms themselves
- Concerning trust in general:
  - TM-systems are most helpful when a peer lacks information about another peer
    - Past personal experience is a more valuable indication of trust than the (possibly) polluted global aggregate

Future Work

- There is a framework paper under authorship
  - Make source-code available to community
  - Encourage others to implement more algorithms
- Analysis of currently implemented TM-Algs
  - Subjective-Logic seems straightforward
  - EigenTrust still undergoing batch tests
- Refinement of efficiency concerns
  - Demonstrate correctness of speed-ups
- Thank you!