TATA CONSULTANCY SERVICES

Experience certainty.

TATA

Secure and Privacy-Aware Searching in a Peer-to-Peer Networks

Jaydip Sen and Arijit Ukil
Innovation Lab, Tata Consultancy Services
Kolkata, INDIA

Objective of the Work

- The term peer-to-peer (P2P) system encompasses a broad set of distributed applications which allow sharing of computer resources by direct exchanges between systems.
- The unstructured P2P systems suffer from a number of issues e.g., fake content distribution, free riding (peers who do not share but consume resources), whitewashing (peers who leave and join the system in order to avoid penalties), search inefficiency and scalability problems and user privacy breach.
- To combat inauthentic file downloads as well as to improve search scalability, this work proposes an adaptive trust-based searching algorithm for P2P networks.
- The novel contribution of the work is that it combines the functionalities of trust management and semantic community formation. The trust management scheme segregates honest peers from the malicious ones, while the semantic communities adapt topology to form cluster of peers sharing similar contents. It also provide a mechanism to protect user privacy in a peer-to-peer network.

Issues in Designing the Proposed Algorithm

- Following design issues are first discussed before the algorithm is presented:
  - Network topology
  - Content distribution model
  - Query initiation model
  - Trust management engine

- Network topology: the network has been modeled as a power law graph, where the degree of the peers follows power law distribution, i.e. fraction of peers having degree L is L^{-k} where k is a network dependent constant.
  - Prior to each simulation cycle a fixed percentage of peers are randomly marked as malicious.
  - The links are categorized into two groups: connectivity and community. The connectivity links are the edges of the original graph. These links are never deleted during the execution of the algorithm.
  - The community edges are added between the peers who have trust between them. A community link may be deleted when the perceived trustworthiness of a peer falls below a threshold.

Issues in Designing the Proposed Algorithm

- Content distribution: each distinct file *fc,r* is abstractly represented by the tuple (*c*, *r*) where *c* represents the content category to which the file belongs and *r* represents the popularity rank within a content category *c*.
- Each peer randomly chooses the content categories to share files and shares more files in more popular categories.

Content distribution among five peers

Peers	Content categories
P1	{C1, C2, C3}
P2	{C2, C4, C6, C7}
P3	{C2, C4, C7, C8}
P4	{C1, C2}
P5	{C1, C5, C6}

C1 is the most popular category and most replicated. P1 shares three categories C1, C2, and C3, where files in C1 categories are more in number than C2 , which in turn, more than C3 category files.

Issues in Designing the Proposed Algorithm

- Query initiation model: the number of queries a peer issues may vary from peer to peer and is modeled as by a Poisson distribution. The probability that a peer issues a query for the file *fc,r* depends on the peer's interest level in category *c* and rank *r* of the file within the category.

- Trust management engine: it enables peers to compute trust metric for other peers.
  - The framework employs a beta distribution for reputation information.
  - The first-hand and second-hand information are combined to compute the reputation value of a peer.
  - The weight assigned by a peer *i* to a second-hand information received from a node *k* is a function of reputation of node *k* as maintained in node *i*.
  - Higher weights are assigned to recent observations. For updating reputation value using second-hand information, Dempster- Shafer theory and belief discounting model may be used.
  - Trust value of a peer lies in the interval [0, 1]. A peer is trustworthy if its trust value is ≥ 0.5 and malicious if its trust value is < 0.5.

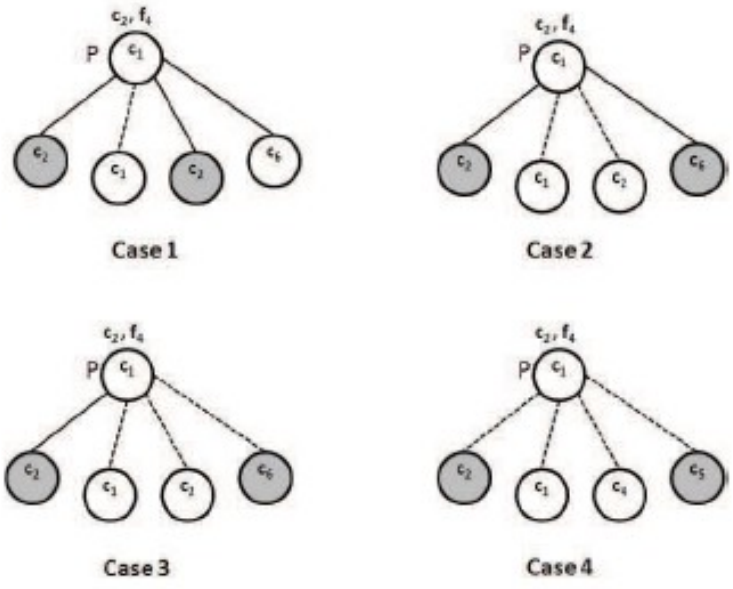
The Proposed Trust-Based Search Algorithm

- The proposed scheme has three steps for its operation:
  - Search
  - Trust verification
  - Topology adaptation

- A TTL bound search is used. At each hop, the query is forwarded to a subset of neighbors which is determined based on the local estimate of network connectivity (Probcom). If a node has a low value of (Probcom), it has more capacity to accept new community edges.
- As the connectivity of good nodes increases, they focus on directing the queries to appropriate community which may host the specific file rather than expanding the community.
- The search is carried out in two steps
  - Query initiation
  - Query forward.

The Proposed Trust-Based Search Algorithm

- Query initiation: The initiating peer forms a query packet containing the name of the file (c, r) and forwards it to some of its neighbors along with the Probcom and TTL values.
- The neighbors are ranked based on trust values and similarity of interest. Preference is given to trusted neighbors who share similar contents.

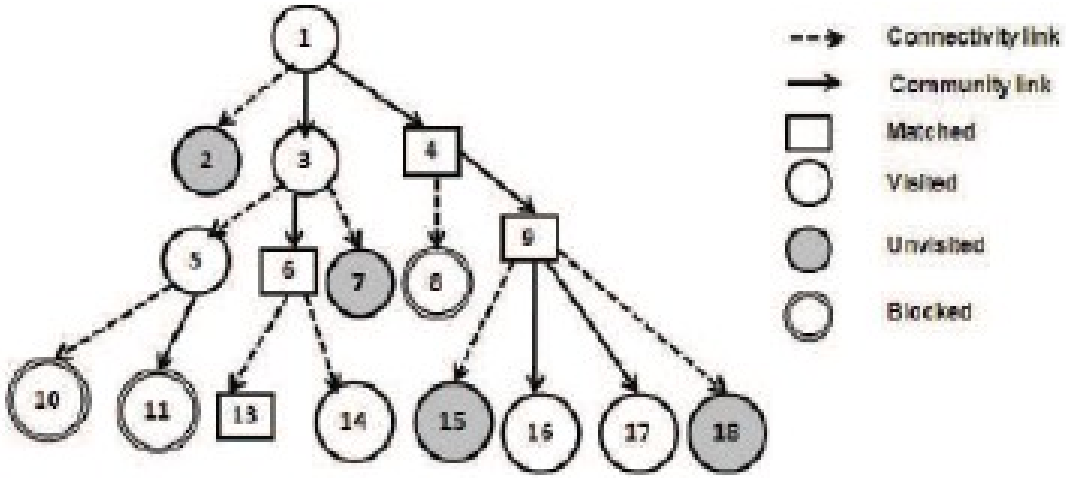


It is assumed that in each case, only two neighbors are selected.

Neighbor selection at P for query string (c2, f4). Community edges and connectivity edges are drawn with solid and dotted lines respectively. Nodes that dispatch query are shaded.

The Proposed Trust-Based Search Algorithm

- Query forward: when a query reaches a peer *i*, it performs the following operations:
  - Checks trust level of peer *j*
  - Checks the availability of the file
  - Calculates the number of messages to be sent
  - Chooses the neighbors



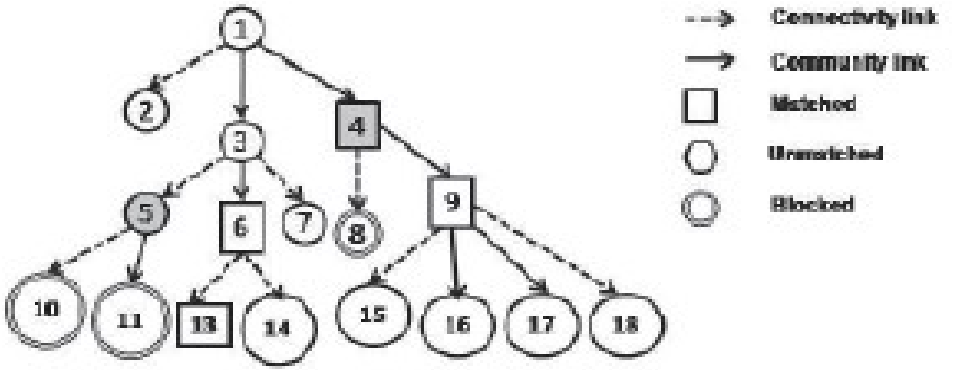
Peer 1 initiates a query and forwards it to two community neighbors 3 and 4. The query reaches peer 8 via peer 4. However, peer 8 knows that peer 4 is malicious. It blocks the query. The query forwarded by peer 5 is blocked at peer 10 and 11 as both know that 5 is malicious. The query is matched at four peers: 4, 6, 9 and 13.

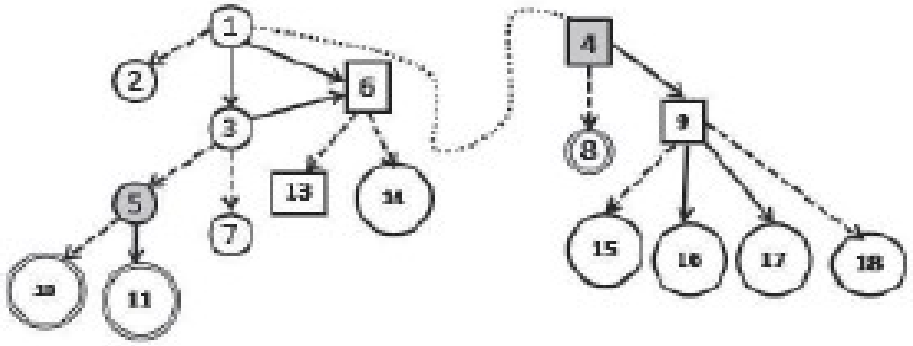
Breadth first search (BFS) tree for search procedure initiated by peer 1

The Proposed Trust-Based Search Algorithm

- Topology adaptation: the responses are sorted by the initiating peer *i* based on the reputation of the resource providers. The peer having the highest reputation is chosen as the source of download.
- If the file provided by peer *j* is found to be spurious, *i* attempts to download it form other sources, and updates the trust rating of *j* and possibly adapts the topology.
- The restructuring of the network is controlled by a parameter called "degree of rewiring". It is the probability with which a link is formed between two peers.
- Topology adaptation consists of the following operations:
  - Link deletion: peer *i* deletes the existing community link with peer *j* if it finds peer *j* as malicious.
  - Link addition: peer *i* probabilistically forms community link with peer *j* if resource is found to be authentic.

The Proposed Trust-Based Search Algorithm





Peer 1 downloads the file from peer 4 and finds it spurious. It reduces the trust rating of 4 and deletes the community link 1 – 4. It then downloads the file from peer 6 and gets an authentic file. Peer 1 now sends a request to peer 6, and the latter grants the request and adds the community edge 1 – 6.

Topology adaptation on the previous breadth first search. Malicious peers are shaded.

Privacy Preservation in Searching

- Identity of the peers: each peer generates a 1024 bit public/private RSA key pair. The public key serves as the identity of the peer. In addition, a distributed hash table (DHT) is maintained that lists the transient IP addresses and port numbers for all peers and for all applications running on the peers.
- Identity of the requesting peer is protected by having the requestor peer ask one of its trusted peer to look up the data on its behalf. Once the data source is identified, the trusted peer acts as a proxy to deliver the data to the requestor. Messages are encrypted by 1024-bit RSA key.
- Protecting the data handle: to improve privacy level, data handle is not put in the request at the beginning. The requestor compute the hash value of the handle and send only a part of the hash results to the trusted peer.
  - Depending on the length of the hash, the receiver peer may find multiple matches. It provides a candidate sets to the trusted peer, which in turn sends the set back to the requestor. A Bloom filter is used for this purpose.
  - Privacy is much improved over the previous case, since an adversary now needs to compromise the trusted node, the Bloom filter and the hash function to attack the privacy.

Privacy Preservation in Searching

- Hiding the data content: each in the previous two scenarios, the privacy of the requestor will be compromised if the trusted node can see the data content when it relays the packets to the requestor.
  - To improve the privacy level and prevent eavesdropping, the data handle and data content may be encrypted
  - If the identity of the supplier is known to the requestor, it can encrypt the request using the supplier's public key. The public key of the requestor, however, cannot be used since the certificate will reveal its identity
  - The requestor generates a symmetric key and encrypts it using the supplier's public key. Only the supplier can recover the key and use it to encrypt the data.
  - To prevent the trusted node of the requestor from conducting a man-in-the-middle attack, the trusted node is required to sign the packet. This provides a non-repudiation evidence and shows that the packet is not generated by the trusted node itself
  - The privacy level has further improved since now the adversary needs to break the encryption keys as well in order to launch an attack

Performance Evaluation

- To analyze the performance of the proposed protocol, three metrics are defined:
  - Attempt ratio (AR): it is the probability that the authentic file is downloaded in the first attempt by a peer. A high value of AR is desirable.
  - Closeness centrality (CC): due to topology adaptation, the length of the shortest path between a pair of good peers decreases. If *Pij* is the length of the shortest path between *i* and *j* through community edges and if *V* denotes the set of peers, then CC for peer *i* is given by:

$$CC_i = \frac{1}{\sum_{j \in V} P_{ij}}$$

  - Largest connected component (LCC): it will be highly probable that the trust-aware overlay graph will be a disconnected graph. LCC is the largest connected component of this disconnected graph. It can be taken as a measure of the goodness of the community structure.

Performance Evaluation

- More metrics….
  - Relative increase in connectivity (RIC): after a successful download, a requesting peer attempts to establish a community edge with the resource provider, if approved by the latter. The metric RIC measures the number of community neighbors a peer gains with respect to its connectivity neighbors in the initial network topology.
  - If *Dinit(i)* and *Dfinal(i)* are the initial and final degrees of peer *i* and *N* is the number of peers, then RIC for peer *i* is computed using the following:

$$RIC(i) = \frac{1}{N} \sum_{i} \frac{D_{final}(i)}{D_{initial}(i)}$$

Performance Evaluation

- More metrics….
  - Clustering coefficient (CLC): it gives an indication about how well the network forms cliques and plays an important role in choosing the TTL value. For higher CLC value, lower TTL values may be used. If *Ki* be the number of community neighbors of peer *i*, then CLC of peer *i* is defined as;

$$CLC_i = \frac{2E_i}{K_i x (K_i - 1)}$$

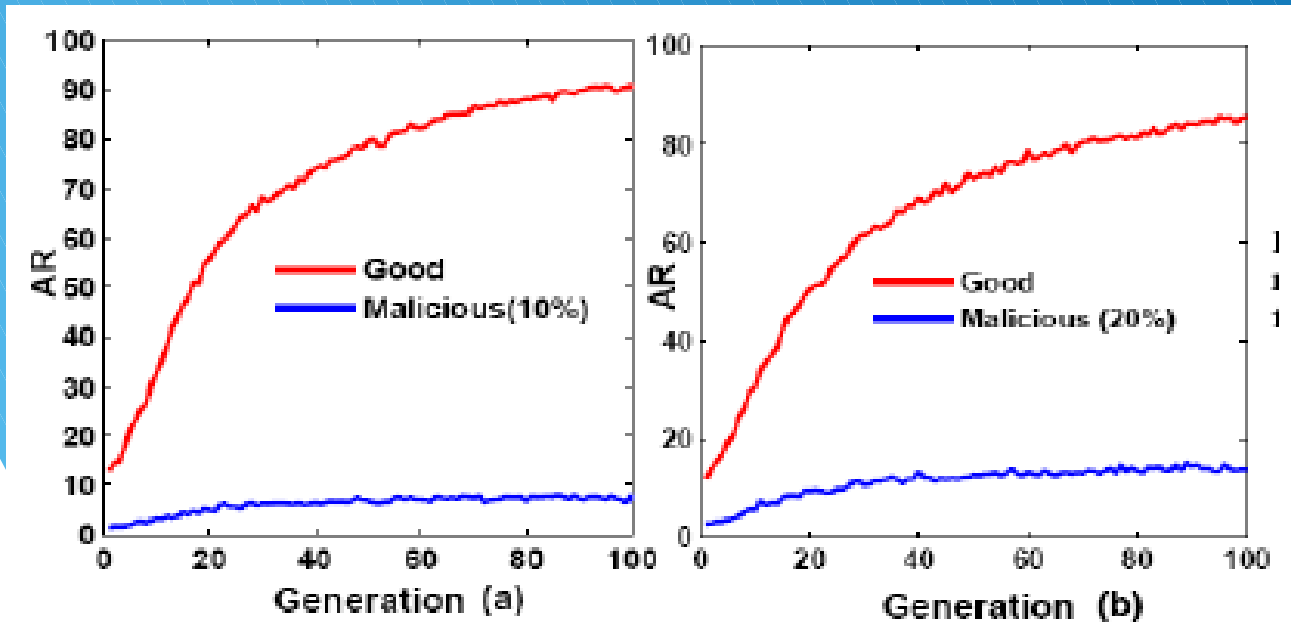
Ei is the actual number of community edges between the *Ki* neighbors. CLC of the network is the average value of all CLC(*i*)s.

  - Trust query propagation overhead (TQPO): it is defined as the total number of distinct DFS search attempts per generation. A trust query may be initiated multiple number of times for a single file search operation : to select a trusted neighbor or to approve a community link.

Performance Evaluation

- A discrete time simulator written in C is used for simulation. In simulation, 6000 peer nodes, 18000 connectivity edges and 32 content categories are chosen.
- Degree of deception (probability of a malicious node providing authentic file) is taken as 0.1.
- Due to formation of semantic edges, the degrees of the peers are allowed to increase maximum upto 30%. In other words, the degree of rewiring is taken as 0.3.
- The TTL value for BFS and DFS are taken as is taken as 5s and 10s respectively. Barabasi-Alabert generator is used to generate initial power law graphs with 6000 nodes and 18000 edges.
- The number of searches per generation and the number of generations per cycle are taken as 5000 and 100 respectively.

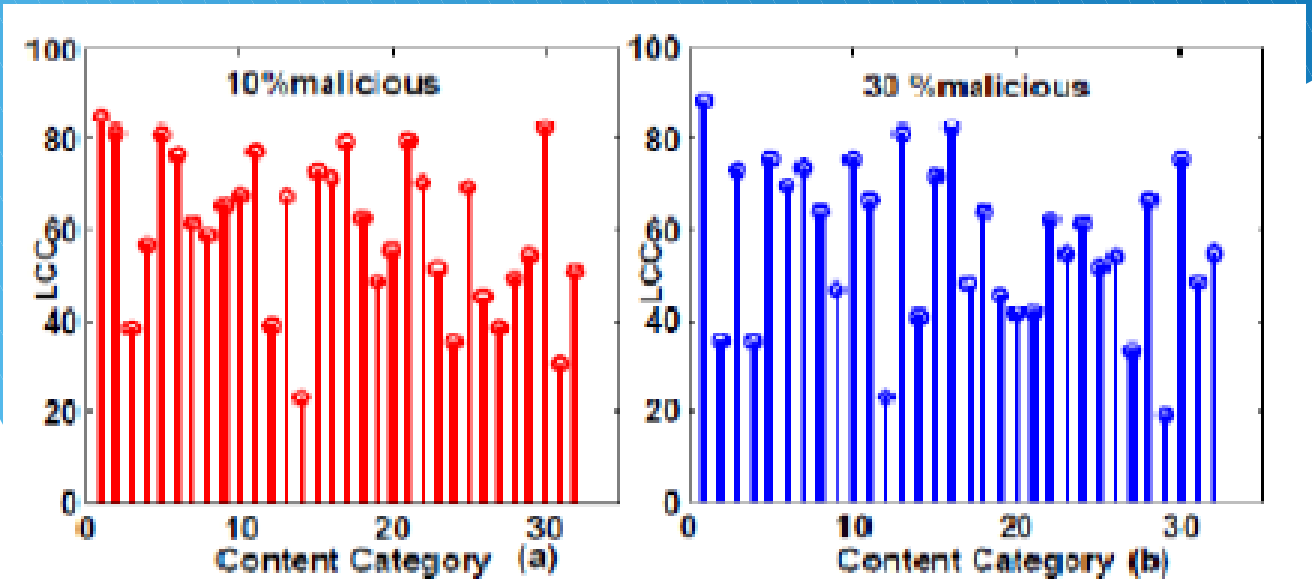
Performance Evaluation



AR vs. percentage of malicious peers. (a) 10% & (b) 20%

Observation: With the increase in the percentage of malicious peers, cost for malicious peers to download authentic files decreases. The reverse is the case for honest peers.

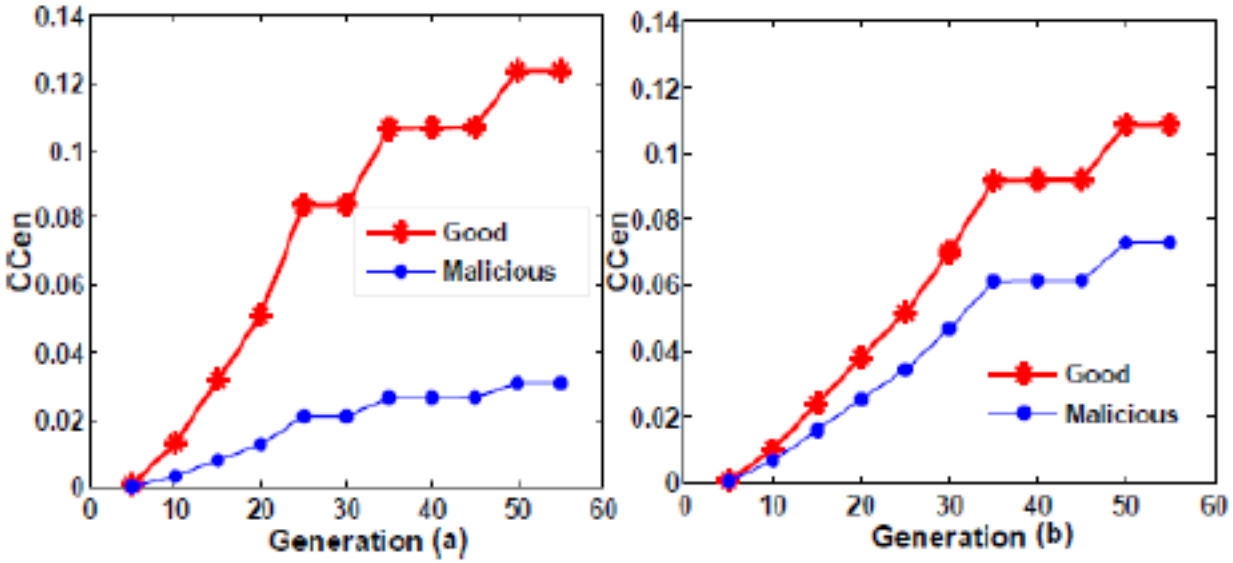
Performance Evaluation



Largest connected components for peers having different content categories

Observation: The average size of LCC for all content categories remains the same even if the percentage of malicious peers increases. It implies that the community formation among the honest peers is not affected by the presence of malicious nodes.

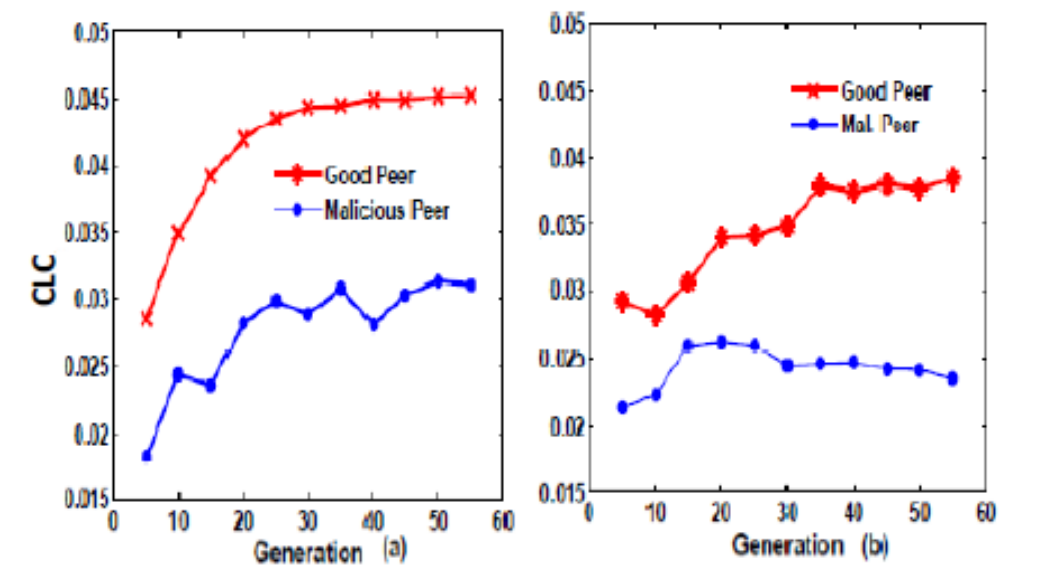
Performance Evaluation



Closeness centrality for (a) 20% & (b) 40% malicious nodes

Observation: The steady state value of CC for honest peers is around 0.12, irrespective of the percentage of malicious peers in the network. For malicious peers, the CC value lies between 0.03 and 0.07. It implies that malicious peers are effectively driven to the fringe of the network while good peers are rewarded.

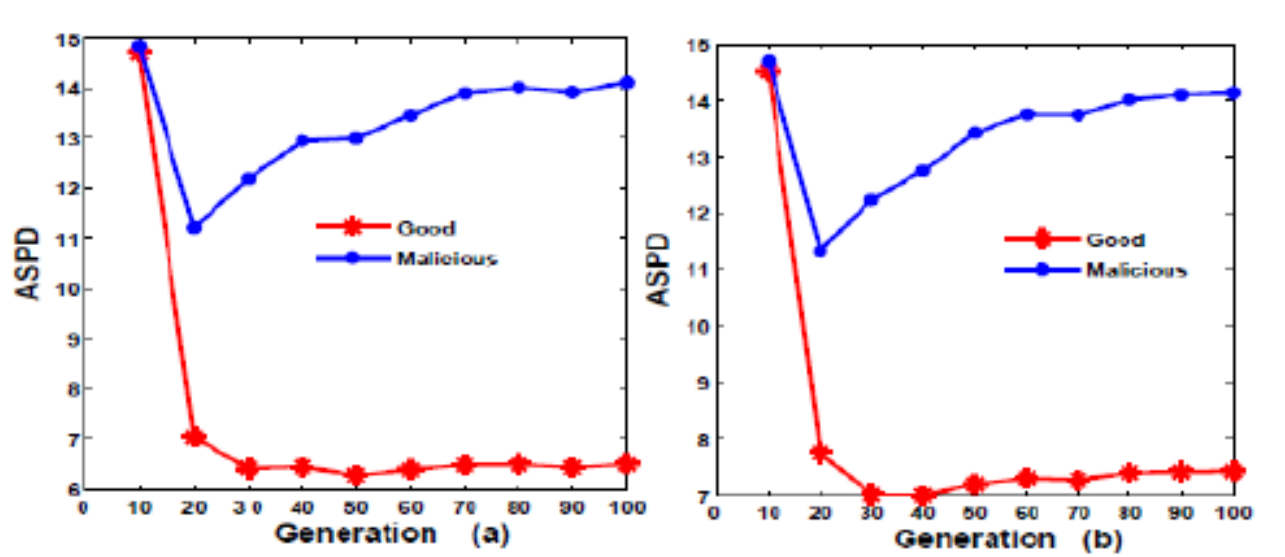
Performance Evaluation



Clustering coefficient for different percentages of malicious peers
(a) 30% and (b) 40% peers malicious

Observation: Clustering coefficients are higher for good peers due to addition of community edges. The clustering coefficients for malicious peers are low due to deletion of community edges. Due to formation of many community edges for the good nodes, a large number of triangles are formed. To get rid of this, search strategy adapts itself from BFS to DFS to minimize redundant message communications.

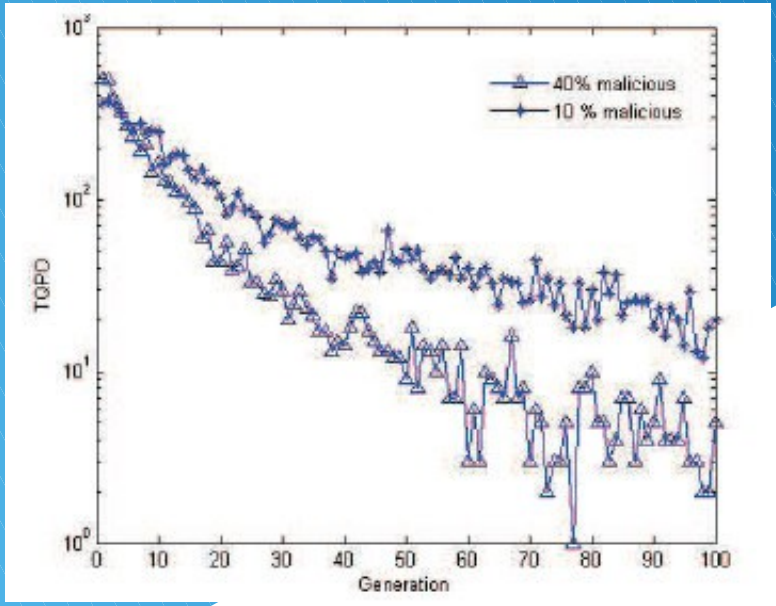
Performance Evaluation



Avg. shortest path length vs. generation of search at the step of ten for different percentages of malicious peers. (a) 30% and (b) 40%

Observation: Good peers have smaller average shortest path length between them. The average shortest path distance decreases for both honest and malicious peers. However, the rate and extent of decrease for honest peers are much higher due to the formation of semantic communities. For malicious peers, ASPD (after an initial fall) increases consistently and finally reaches a max value of 15. The avg. value of ASPD for honest peers is around 6.

Performance Evaluation



Overhead of trust query propagation for different percentages of malicious peers. (a) 30% and (b) 40% peers malicious

Observation: The steady state value of TQPO attains a low value – less than 10 when 10% of the peers are malicious. Even with 40% malicious nodes, TQPO gradually decrease and becomes less than 20 within 100 generations. Trust propagation has little impact on the system overhead since trust information gets embedded in the trust-aware network topology

Conclusion

- A mechanism has been presented that solves multiple problems in peer-to-peer networks, e.g., spurious file downloads, poor search scalability, free riding, whitewashing and user privacy breach.
- It has been shown that by topology adaptation, it is possible to isolate the malicious peers while providing topologically advantageous positions to the good peers so that good peers get faster and authentic responses to their queries.
- Simulation results have demonstrated that the protocol is robust and efficient even in presence of a large number of malicious peers in the network.
- Analysis of the overhead of the trust management engine and a formal security analysis of the trust framework are two possible future plans of work.

Thank You