Denial of Service Resilience in Peer to Peer File Sharing Systems

D. Dumitriu, E. Knightly, A. Kuzmanovic, I. Stoica,

W. Zwaenepoel

Presented by: Ahmet Canik

Outline

1. Background on P2P systems
2. File targeted DoS attacks
3. Network targeted DoS attacks
4. Modeling resilience to network targeted attacks
5. Simulation study
6. Conclusions

Background on P2P systems

- Can be classified as **structured** or **unstructured** based on whether there is any **inherent structure** in the system **for locating files**.
- Unstructured
  - Gnutella, KaZaA, Freenet, …
- Structured
  - CAN, Chord, Pastry, Tapestry, Kademlia, …
- Hybrid
  - Structella

Gnutella

- A file can be stored at any node in the system
- Two-level hierarchy: **leaf nodes**, **supernodes**
- Each leaf node is connected to one or more supernodes. Supernode maintains a **directory** of all files stored at its leaf nodes.
- Leaf node queries a file from its supernode. If the supernode knows the location of a file copy, it sends the answer back to the requester. Otherwise, the supernode **floods** the query to other supernodes.

Structella

- Each file is associated with a unique ID (e.g. hash value of the file name or content)
- A file is stored at the node responsible for the file's ID. Each node is responsible for a chunk of the ID space.
- Can find the node responsible for a given ID by contacting only $O(\log N)$ nodes.
- Structella is hybrid proposal based on Pastry. Like the original Gnutella, Structella uses flooding to locate files, but does so in a more efficient way.

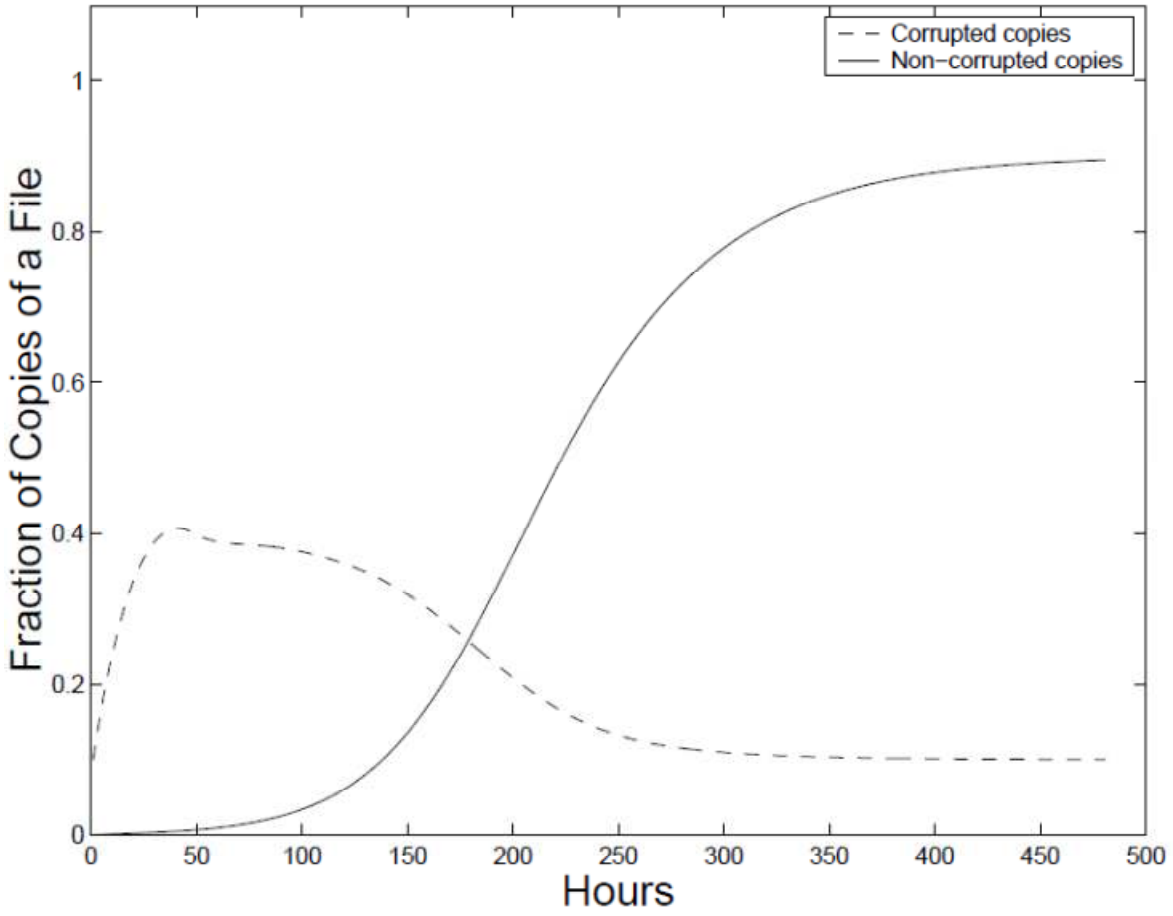
File targeted DoS attacks

- A malicious node advertises a corrupted file, and distributes this copy if it is chosen by another peer.
- P2P topology does not play a role in the effectiveness of a file-targeted attack.
- Instead, the user-behavior factors determine the spread of polluted files.
  - Willingness to share files
  - Speediness in removing corrupted files
  - Persistence in downloading files under attack

Spreading the Pollution

- $M = 15{,}000$ interested nodes
- $b_0 = 1{,}500$ malicious nodes
- $g_0 = 10$ initial good copies
- $s = \frac{1}{24}$ interest-rate factor: each peer interested in obtaining this file attempts to download it on average once per 24 hours
- $L = 48$ a polluted copy can remain at most 48 hours on a user's machine

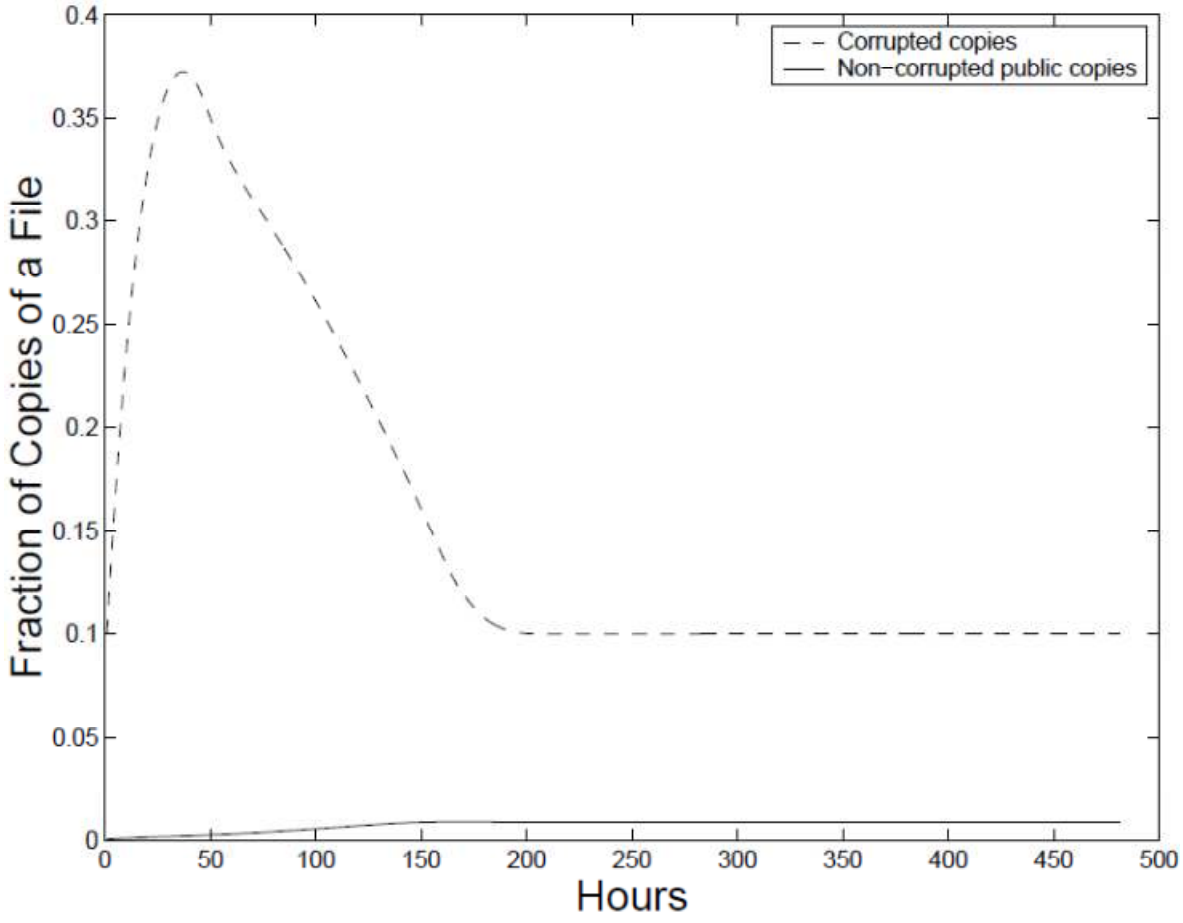
Spreading corrupted and non-corrupted copies



Cooperation and Persistence

- Two fundamental reasons that prevent files targeted by the pollution attack from spreading in the network:
  - Not all peers are willing to share the files that they download
  - A user's interest for downloading newly released audio/video files quickly decreases
- Same parameters with the previous model. Additionally:
  - $1 - p_s = 0.6$ probability that a user is willing to share the file
  - $s_i = \frac{1}{24}\left(1 - \frac{i}{24}0.15\right)$ on average, 15% of users give up after the first day, another 15% after the second day, and so on

The impact of users' greediness and persistence



Network targeted DoS attacks

- Differences from file-targeted (pollution) attacks:
  - In network-targeted attacks, an attacker responds to **all** queries, whereas in the pollution attack it only replies to queries for a **set of targeted files** that are being protected
  - The attacker is able to **intercept** a query for a downstream node and falsify the reply on the reverse path. Hence, a query that follows a path with even a single malicious node gets a response pointing to a bogus file.

System Model

- **Query:** Client queries the system for a particular file and the system returns a number of replies.
  - IP address of the node storing a copy of the queried file
  - Information to calculate the estimated download time, e.g., the node's queue length (ideally including file sizes), the maximum number of simultaneous uploads, and the access link bandwidth
- **Download:** Client selects a node among the nodes contained in the replies it has received, and contacts that node to download

Attacker Strategy

- Receiving any **query** => forwards it normally.
- Requested to forward any **reply** => modifies the reply with false information.
- Assuming that the attacker cannot respond to queries directly, but rather must wait for legitimate replies from downstream in order to modify them.
- Assuming that attackers cannot modify the query forwarding algorithm executed by a legitimate node. Thus, a query that follows a path consisting only of legitimate nodes always generates a correct reply.
- Does not handle **routing protocol attacks**.

False reply attack

- The attacker falsifies the reply by **replacing the replying peer's identity with its own** and by advertising a **very low expected transfer delay**.
- This strategy allows the attacker to respond to requests for files for which it has no or limited information (*e.g.,* the attacker does not know the exact file name).
- If selected by the client, the node transfers a **corrupted file**.

Slow node attack

- The attacker points the client to a non-malicious but **low-bandwidth peer**, and lies about that peer's capabilities, *i.e.,* it changes the advertised delay of slow nodes.
- The attacker also drops replies from fast nodes.

Client Strategy

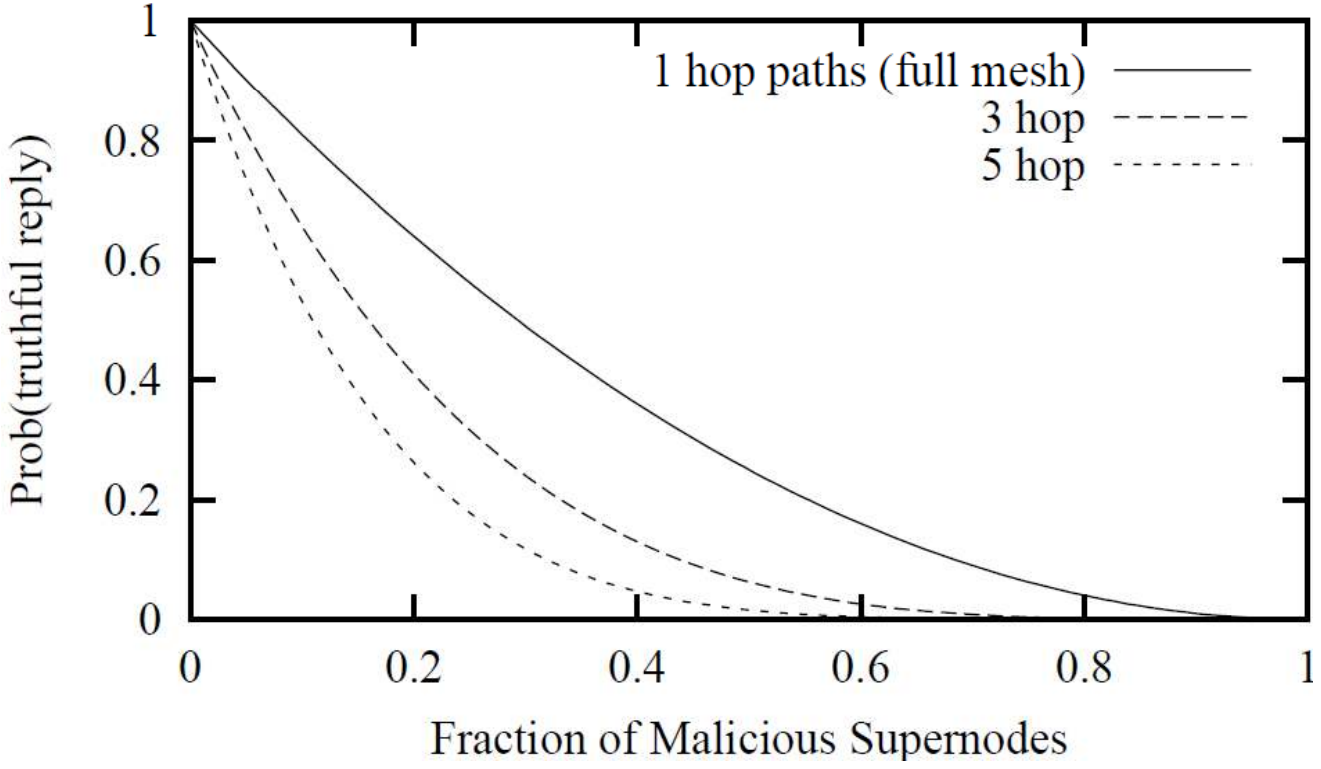
- In response to a query, a client receives a **set of replies** pointing to different nodes. But **which**?
- **Best:** The node that advertises the best performance, *i.e.,* the lowest estimated delay
- **Random:** A random node, independent of the nodes' advertised resources
- **Redundant best:** Redundant downloads from C nodes with the lowest estimated delay. Once the first download finishes and the content is verified for correctness, the other downloads are stopped.

Client Strategy

- **Redundant random:** Redundant downloads from C peers, but chooses those C peers randomly.
- **File Chunking:** The file is sliced into P chunks, and the client downloads a chunk from each of P different peers in parallel
- **Reputation Systems:** A simplified model to mark peers as malicious or non-malicious.
- **Detection:** For the download of a complete file, it is assumed that the client can detect whether a file is corrupted only after it has downloaded the entire file.

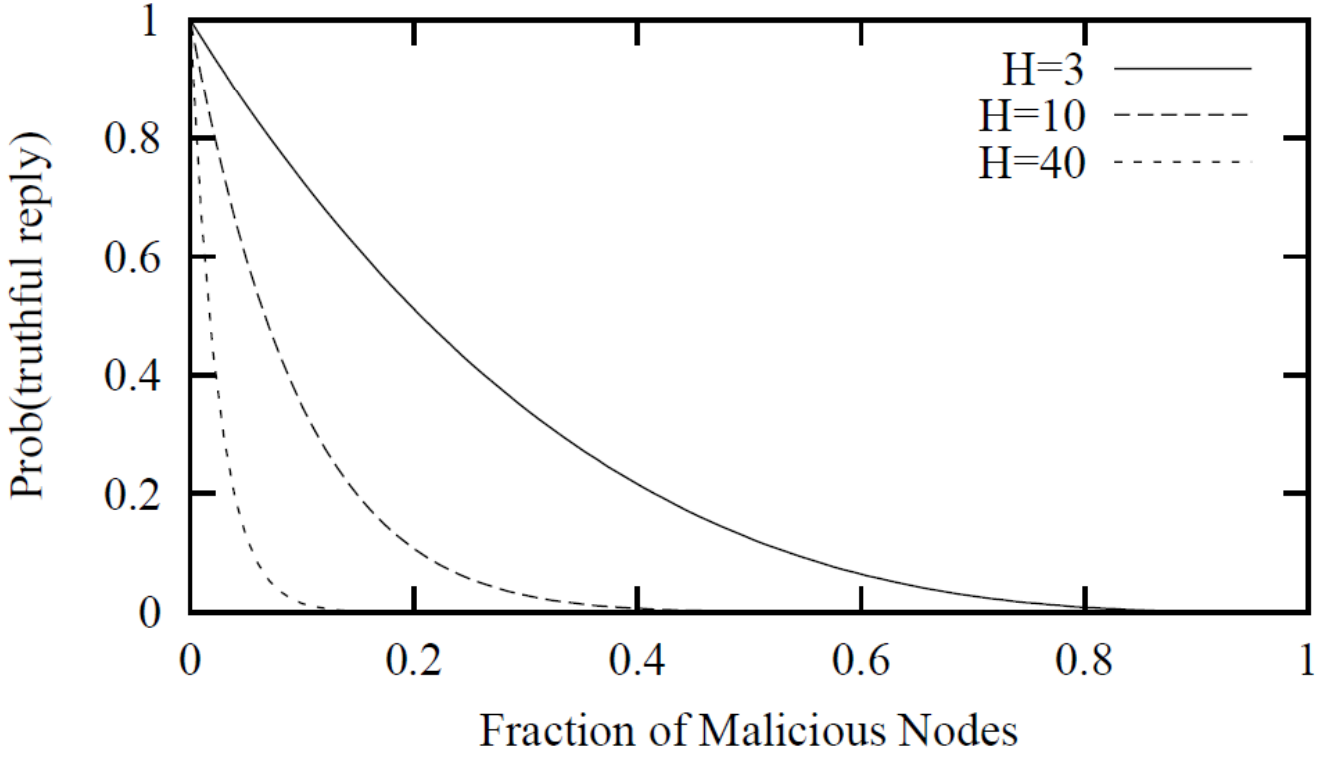
Supernodes and Hierarchy

- Requests and replies are routed via an inter-connected mesh of supernodes. Supernodes reply to queries on behalf of their leaf nodes.



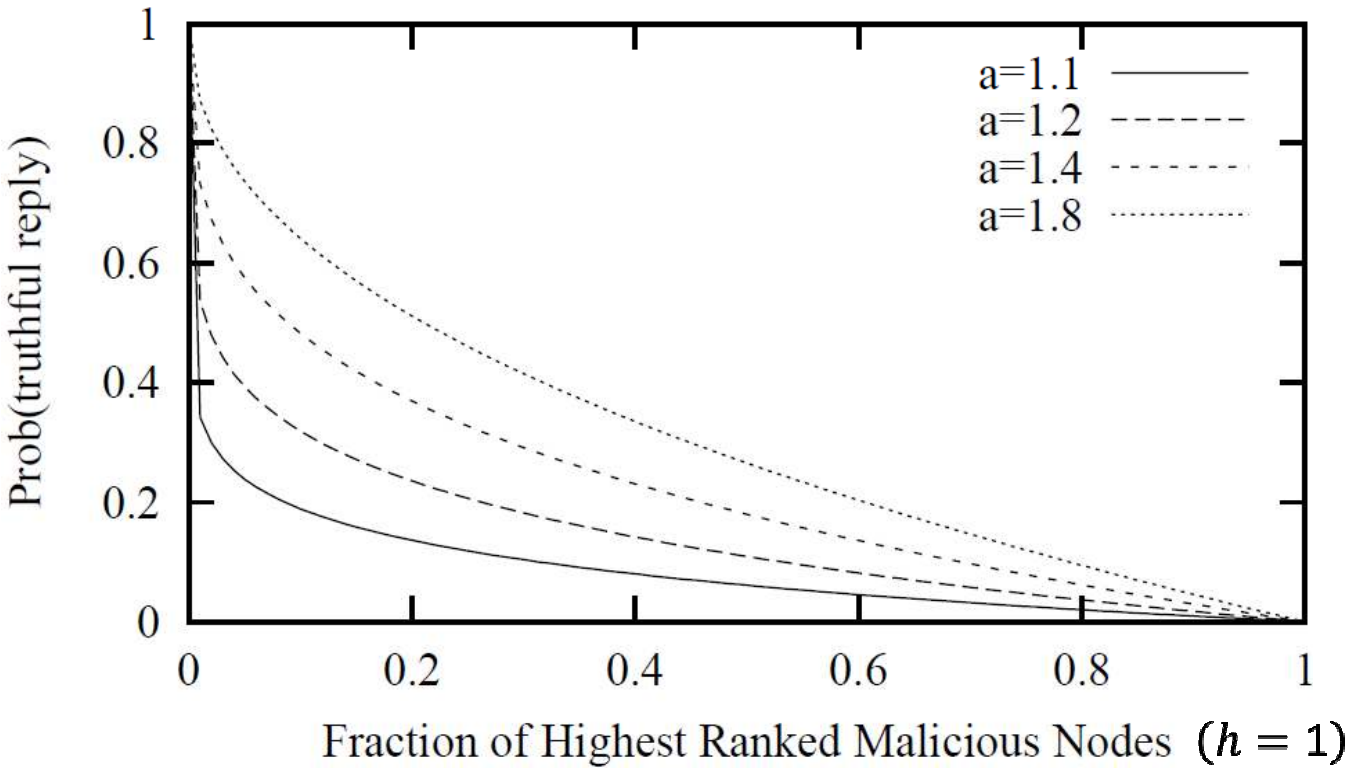
k-Regular Topologies and Path Length

- Structural topology approximated by a k-regular graph, where k is usually $O(\log N)$



Power Law Topologies

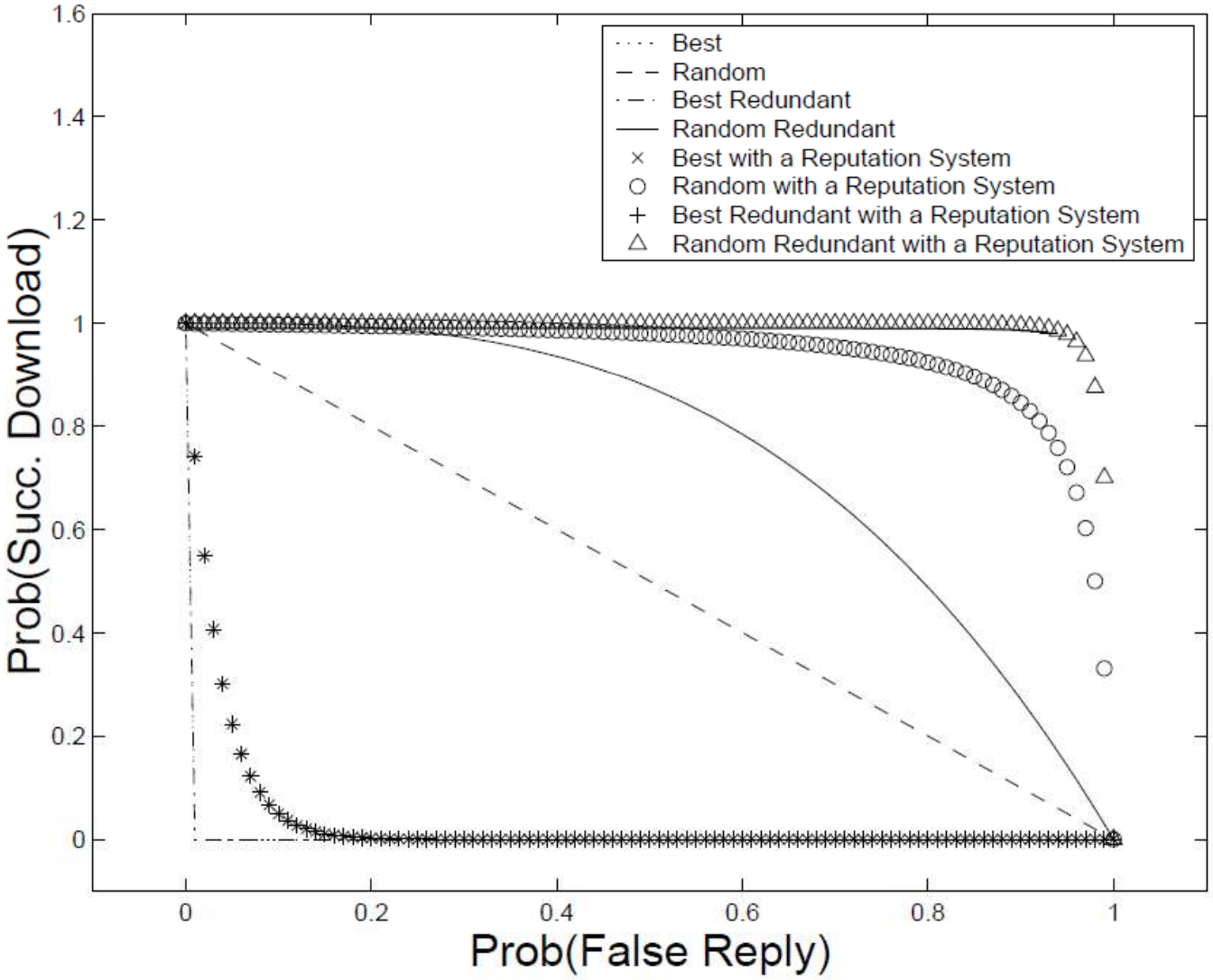
- Participating attacks can be far more devastating.



Client Strategies

- Reputation Systems
  - f_N: The **false-negative** probability is defined as the fraction of malicious nodes that are left undetected by the reputation system
  - f_P: The **false-positive** probability is the fraction of non-malicious users that are falsely declared malicious.
- In the following figure $f_N = f_P = 0.02$ is assumed.
- Reputation systems are **unable to improve** the performance of the "best" strategies.

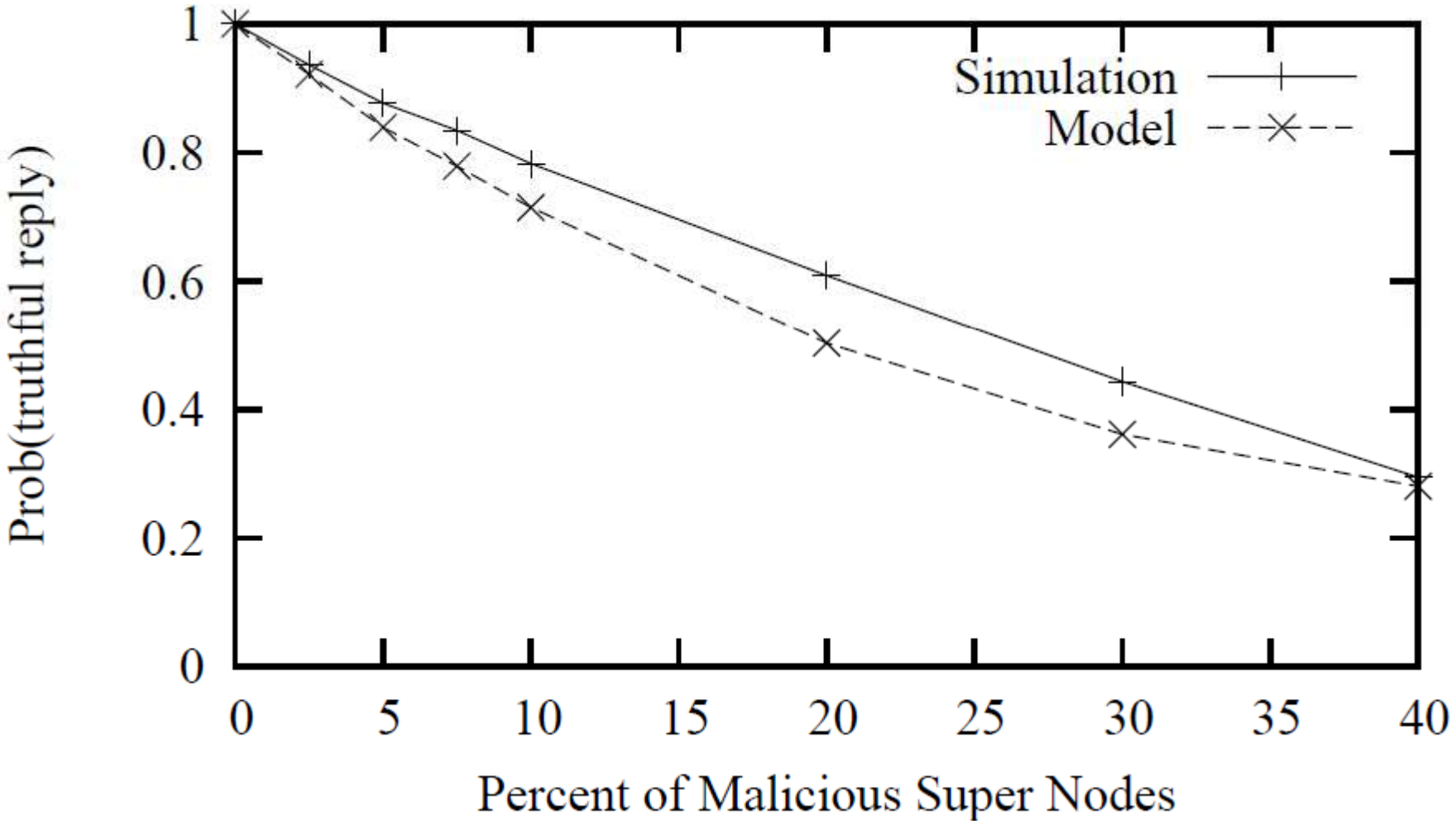
Client Strategies



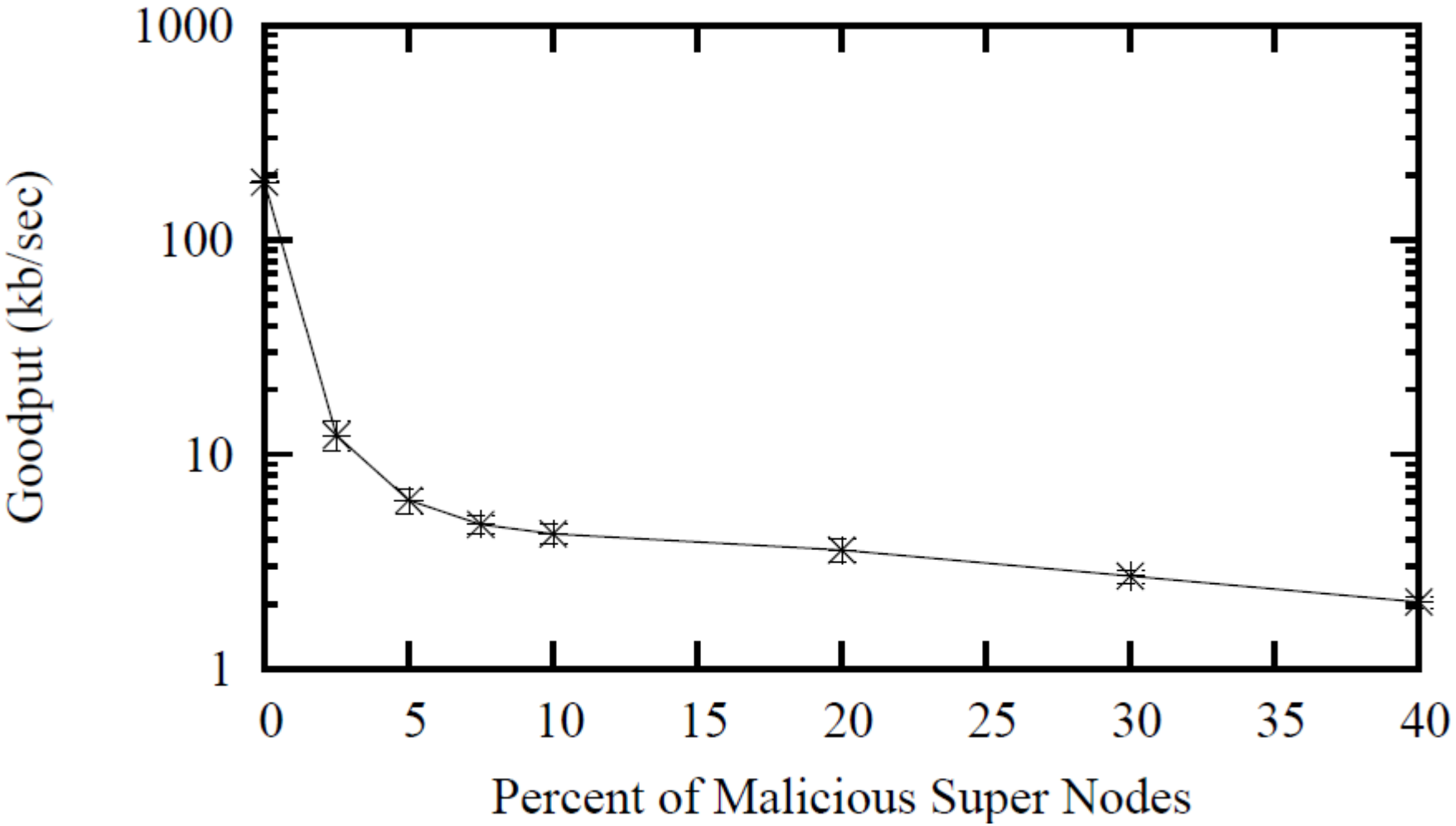
Simulation Preliminaries

- Unstructured: Gnutella
- Structured: Structella over FreePastry
- Leaf nodes: 56 kb/sec – 1 Mb/sec
- Supernodes: 1 Mb/sec – 10 Mb/sec
- Number of nodes = 10,000
- Each scenario is simulated 10 times, averages taken
- Key performance measures
  - probability of truthful reply
  - average system goodput

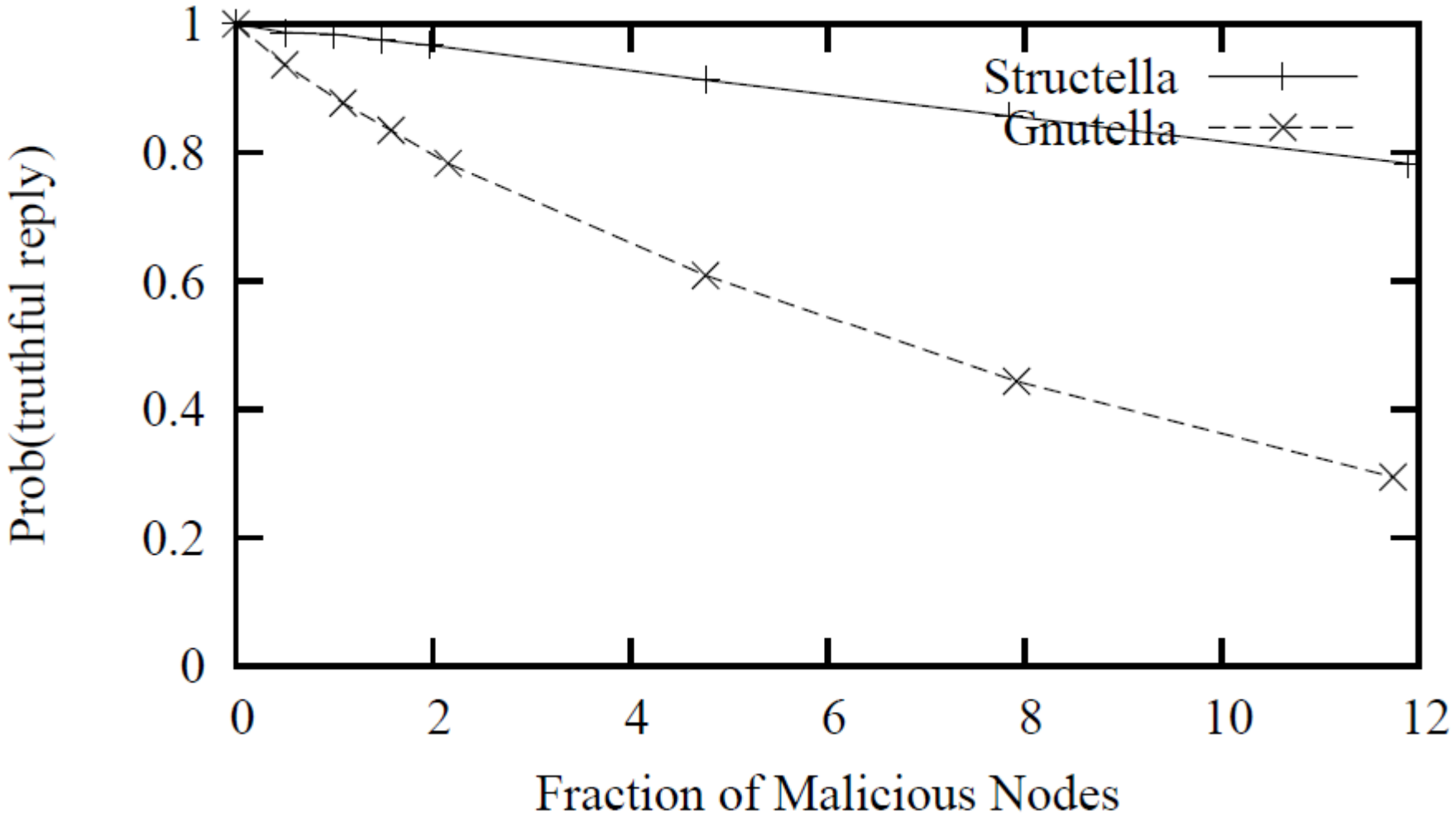
Baseline Attack (Gnutella)



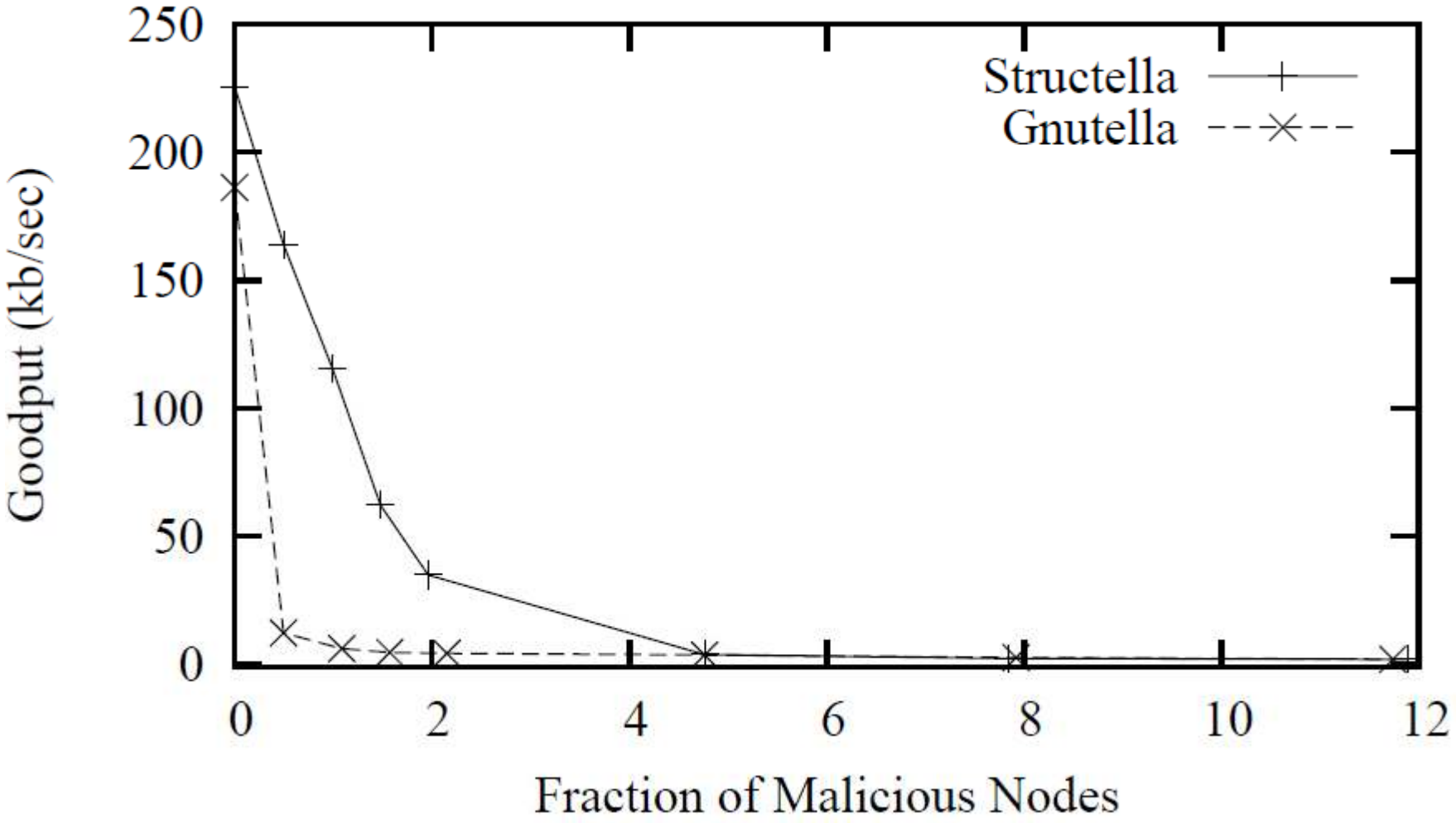
Baseline Attack (Gnutella)



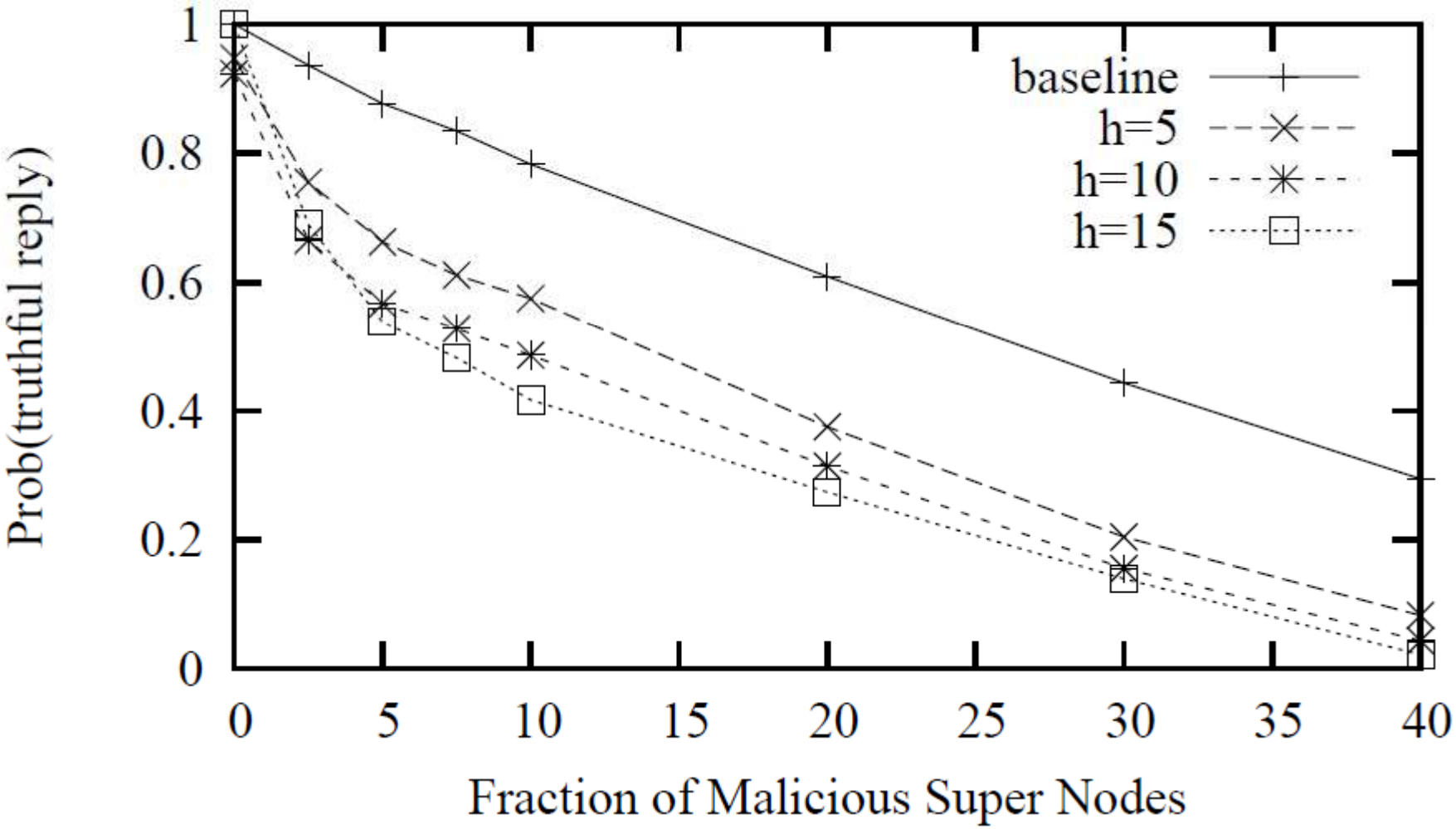
Overlay Structure & Hierarchy



Overlay Structure & Hierarchy



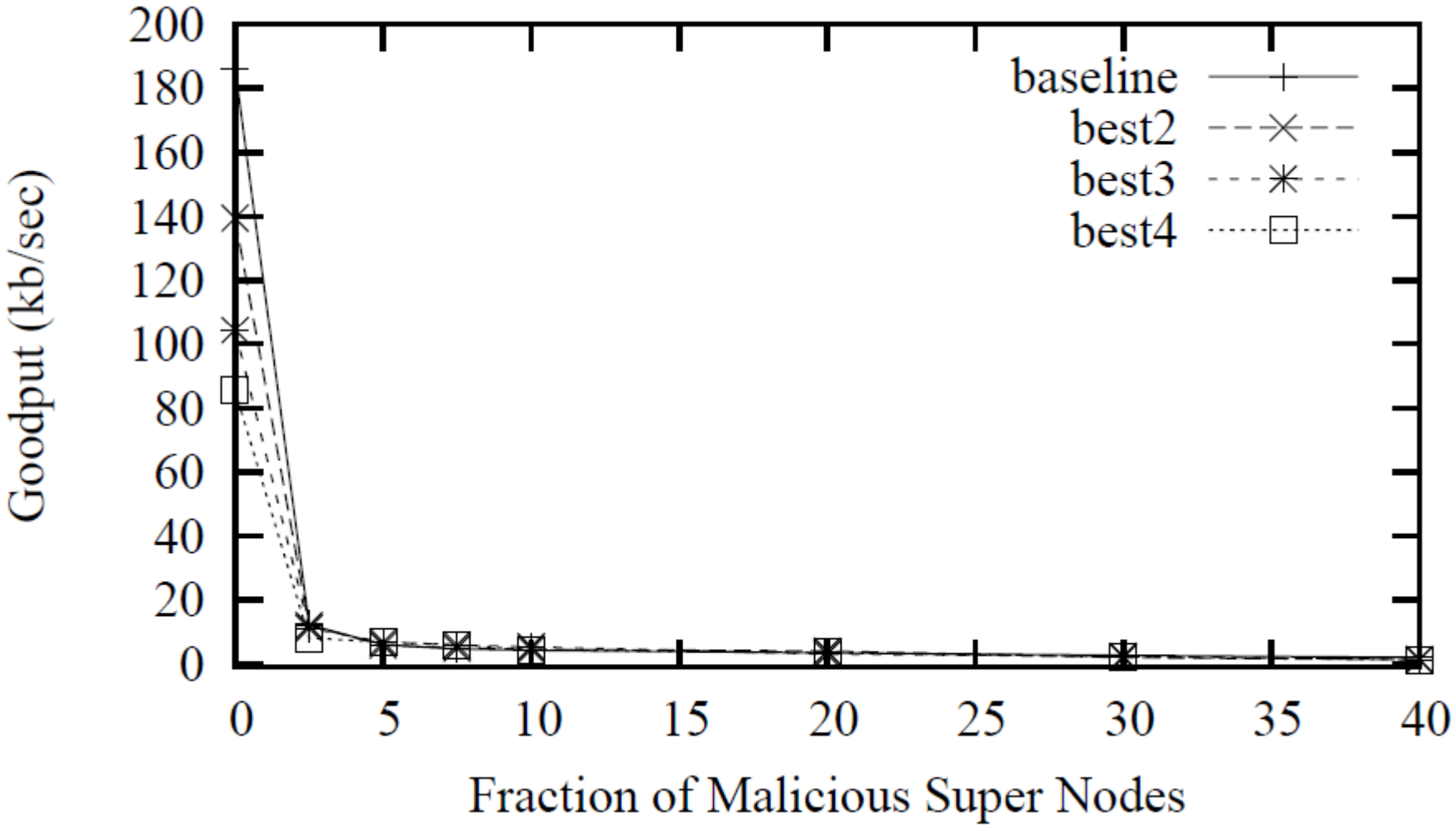
Path Length


Prob(truthful reply)
1
0.8
0.6
0.4
0.2
0
0
5
10
15
20
25
30
35
40
Fraction of Malicious Super Nodes
baseline
h=5
h=10
h=15

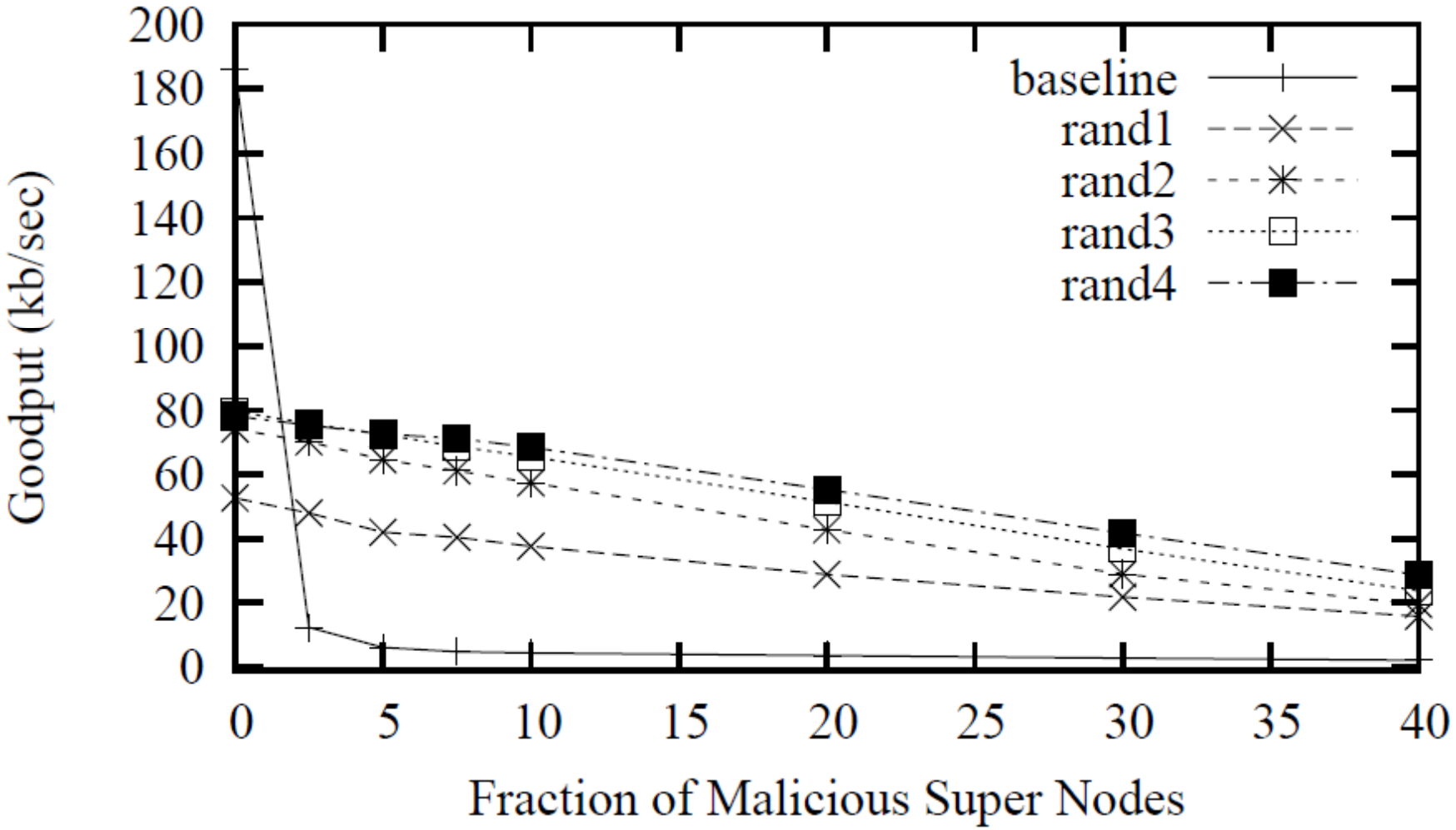
Victim Counter Strategies

- Users do not sit by idly when the system is under attack.
- They use **trial and error** to find effective counter-DoS strategies to improve their performance.
- Users may invoke **multiple downloads** in order to decrease their own delay, perhaps without consideration of adverse effects on others' performance.

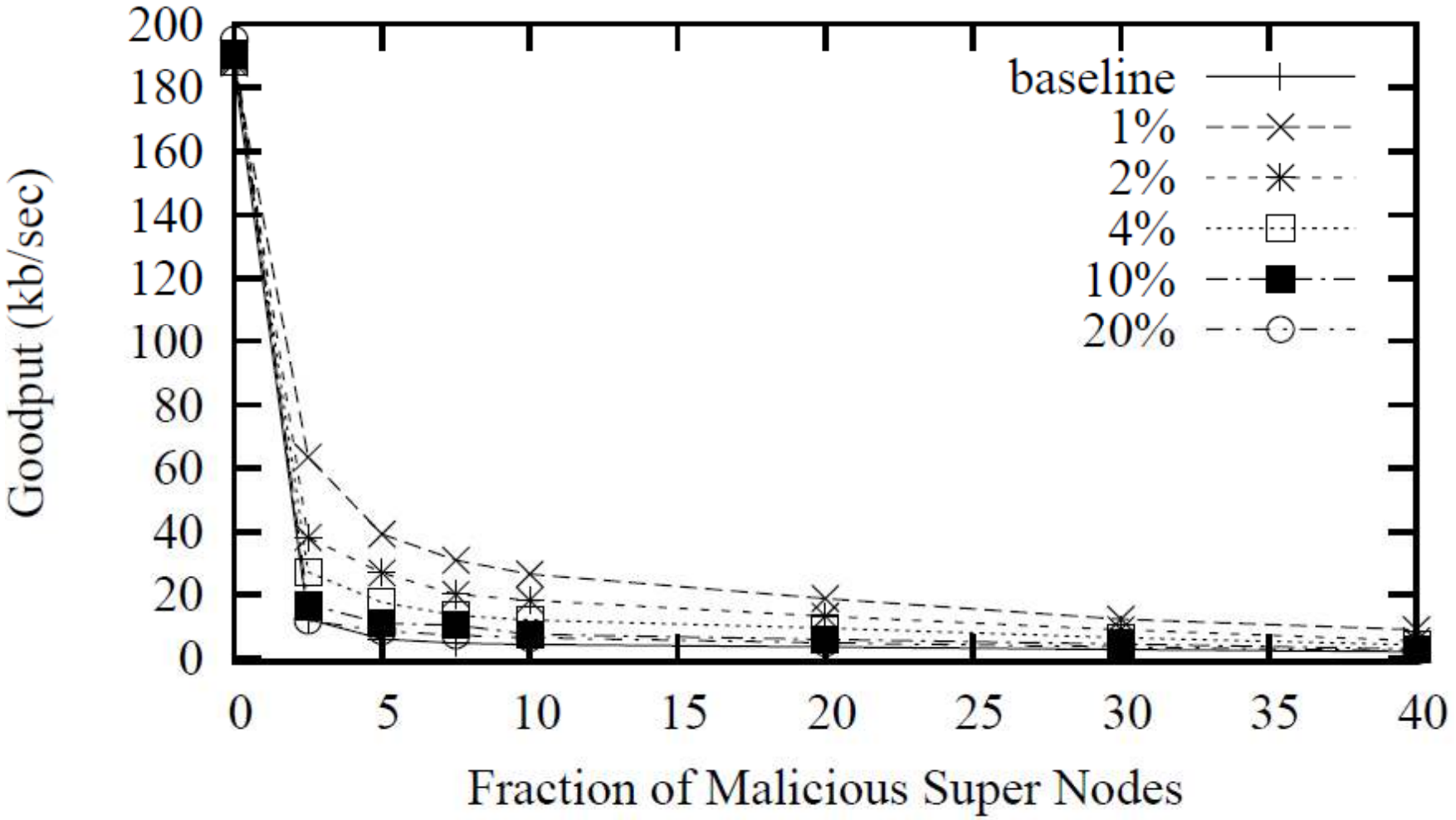
Best N Redundant Download



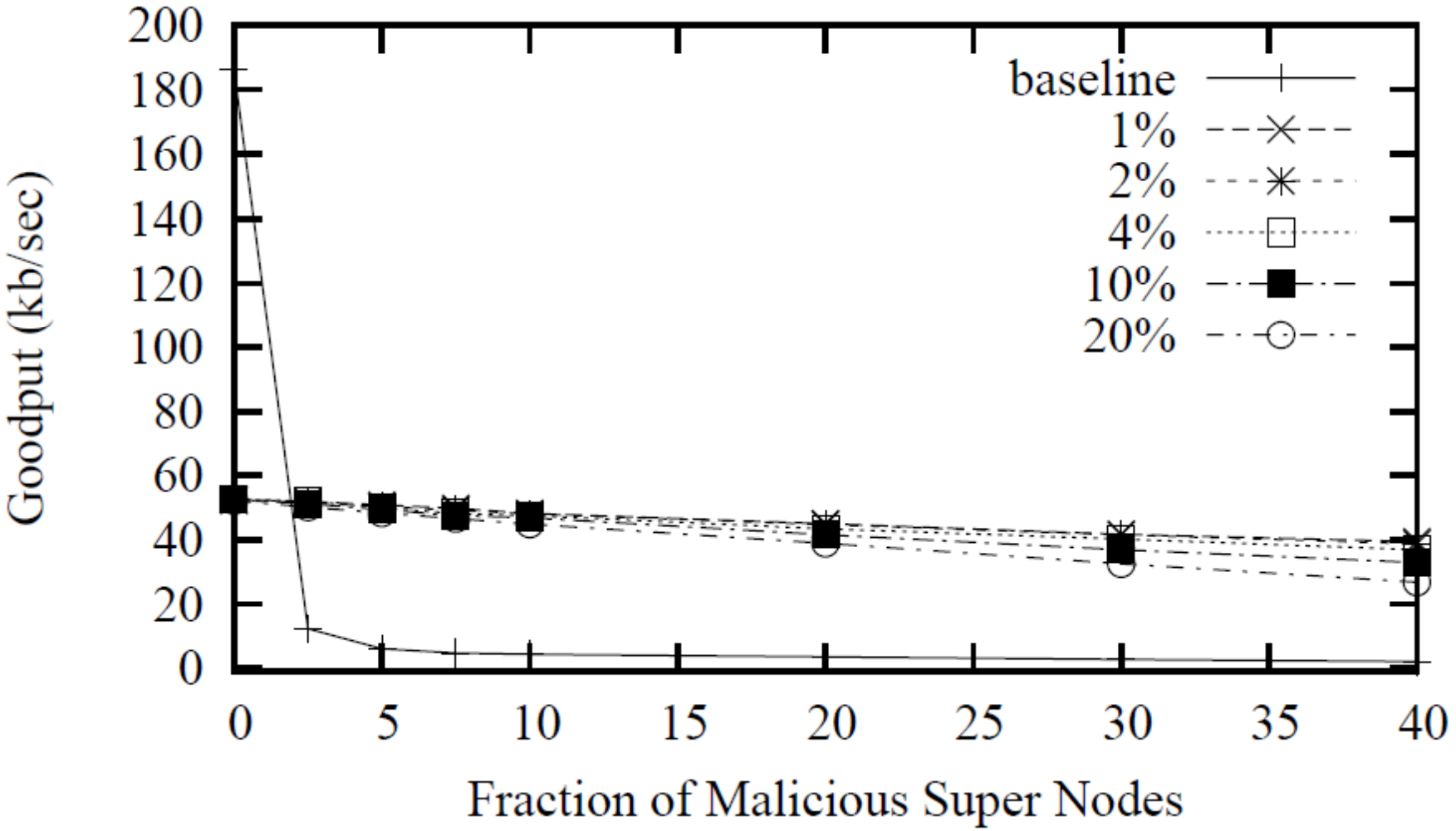
Random Redundant Download



Reputation Systems & Best



Reputation Systems & Random



Conclusions

- File-targeted (pollution) attacks applied are largely **inefficient** in cooperative P2P environments due to **scalability** limitations. The main reasons for their current success:
  - Clients do not share files
  - Clients do not remove corrupted files
  - Clients quickly give up when the system is under attack
- Structured P2P systems are **more resilient** than hierarchical P2P systems as the additional protocol functionality of nodes in the first-level of the hierarchy provides an **acute** DoS vulnerability.

Conclusions

- In both cases, system goodput degrades tremendously (**hyperexponentially** fast) with the number of malicious nodes, when users select to download files from the peer with **best-advertised** download time.
- **Reputation** systems are largely **ineffective**, even with a very small number of false negatives.
- Randomization techniques are able to transform the system's resilience from a hyperexponential scaling to a more **linear** scaling. Unfortunately, randomization severely hinders performance when no attackers are present.

References

- Dumitriu, D., Knightly, E., Kuzmanovic, A., Stoica, I., & Zwaenepoel, W. (2005, June). Denial-of-service resilience in peer-to-peer file sharing systems. In *ACM SIGMETRICS Performance Evaluation Review* (Vol. 33, No. 1, pp. 38-49). ACM.

Thanks

- Questions & Answers