

Attacking P2P Networks

Scott G. Miller & Oskar Sandberg
FreenetProject Inc.

Introduction

- File sharing and publishing networks are one of the most important uses for P2P.
- Networks that survive "in the wild" need to be resistant to attacks, especially denial of service and data removal.
- Most current system are vulnerable.

Freenet Project

Aims to build an Internet wide highly survivable and decentralized file publishing network.

Software:

Fred - Reference implementation of a Freenet routing node.

Serapis - Flexible network simulator

Simulated Topologies

- Freenet routing.
- "Hypermesh" (Plaxton variant) routing.
- Broadcast (Gnutella) querying.
- Others.

Freenet Routing

Freenet routes using a probabilistic model with active feedback.

Nodes learn which neighbors are best at providing data within a region of the "keyspace".

All nodes cache data aggressively, and data has no permanent home, ie it resides only where currently cached.

Freenet Routing Pros

- Easy to understand and implement.
- Easy and painless for Nodes to join and leave the network.
- Gives good search performance under right conditions.
- It is difficult difficult to try to predict and map the network ("Heisenberg" quality).

Freenet Routing Cons

- Cannot guarantee successful searches.
- For good performance, a large amount of redundant storage (8-10x) is needed.
- Not extensively studied.

Freenet Routing

For more information see:

*Ian Clarke, Theodore W. Hong, Scott G. Miller, Oskar Sandberg,
Brandon Wiley*

"Protecting Free Expression Online with Freenet"
Design Issues in Anonymity and Unobservability 2000: 46-66

Hypermesh routing

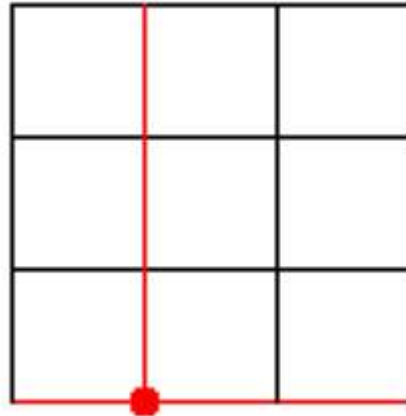
"Hypermesh" is my name for overlay networks that generalize on the idea of routing in a hypercube.

"Plaxton variant" after paper by Plaxton et al.

Examples include Plaxton's model, *Tapestry* (Oceanstore system), *Pastry* (PAST system). *Chord* is very similar.

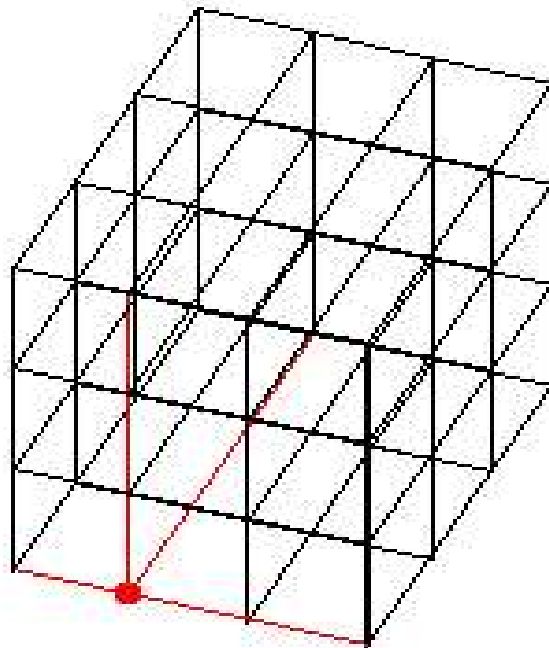
Hypermesh Routing

Nodes are placed in a rectangular mesh, each node knows about those that differ in exactly one coordinate:



Hypermesh Routing

As the network grows, rather than growing the rectangular mesh, add more dimensions:



Hypermesh routing

Data is then placed and located by breaking the binary key value into coordinates and matching with a node.

This is very simplified, in practice the mesh is convoluted to allow greater freedom in choosing neighbors. It is also made to work even if the grid is not completely dense.

Hypermesh Routing Pros

- Guaranteed logarithmic searchtimes.
- Network is exhaustively searched.
- Model is mathematically elegant.

Hypermesh Routing Cons

- Adding new nodes is non-trivial.
- Fickle nature of Internet peers upsets model.
- The network structure is easily manipulated.

Hypermesh Routing

For more, see:

C. Plaxton, R. Rajaraman, A Richa.

Accessing Nearby copies of replicated objects in a distributed environment.

In Proc. of ACM SPAA, June 1997.

B Zhao, J Kubiatawicz, A Joseph.

Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing.

EECS Report No. UCB/CSD-01-1141, Univeristy of California, Berkley, April 2001.

A Rowstron, P Druschel.

Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems.

Microsoft Research, submission to ACM HOTOS VIII, 2001.

Broadcast Routing

- Simplest conceptual style
- Peers pass each query on to all neighbors (with limits)
- Limits to deal with traffic create search 'horizons'.

Broadcast Pros

- Easy to visualize, Easy to implement
- Search accuracy limited only by the horizon
- Lends well to specialized queries (searching)

Broadcast Cons

- Inefficient use of network (many messages sent and nodes contacted to find data)
- Search accuracy limited by a horizon
- Lends well to specialized query abuse (spam)

Others

- Global index.
- "Super Node" type concentrated topology.
- Power law random walks as presented by Adamic et al.

Serapis features:

- Low level simulation (actual protocols)
- Scripting of usage scenarios similar to our experience with Freenet
- Able to simulate underlying network conditions.

Serapis

- The ancient Egyptian god of dreams and chaos.
- Our network simulator.
- Written in java, based on *Gamora* system.
- Intended to test ideas and attacks on Freenet.
- Flexible message passing and scripting system.

Simulations

"Simulations are like miniskirts, they show a lot and
hide the essentials."

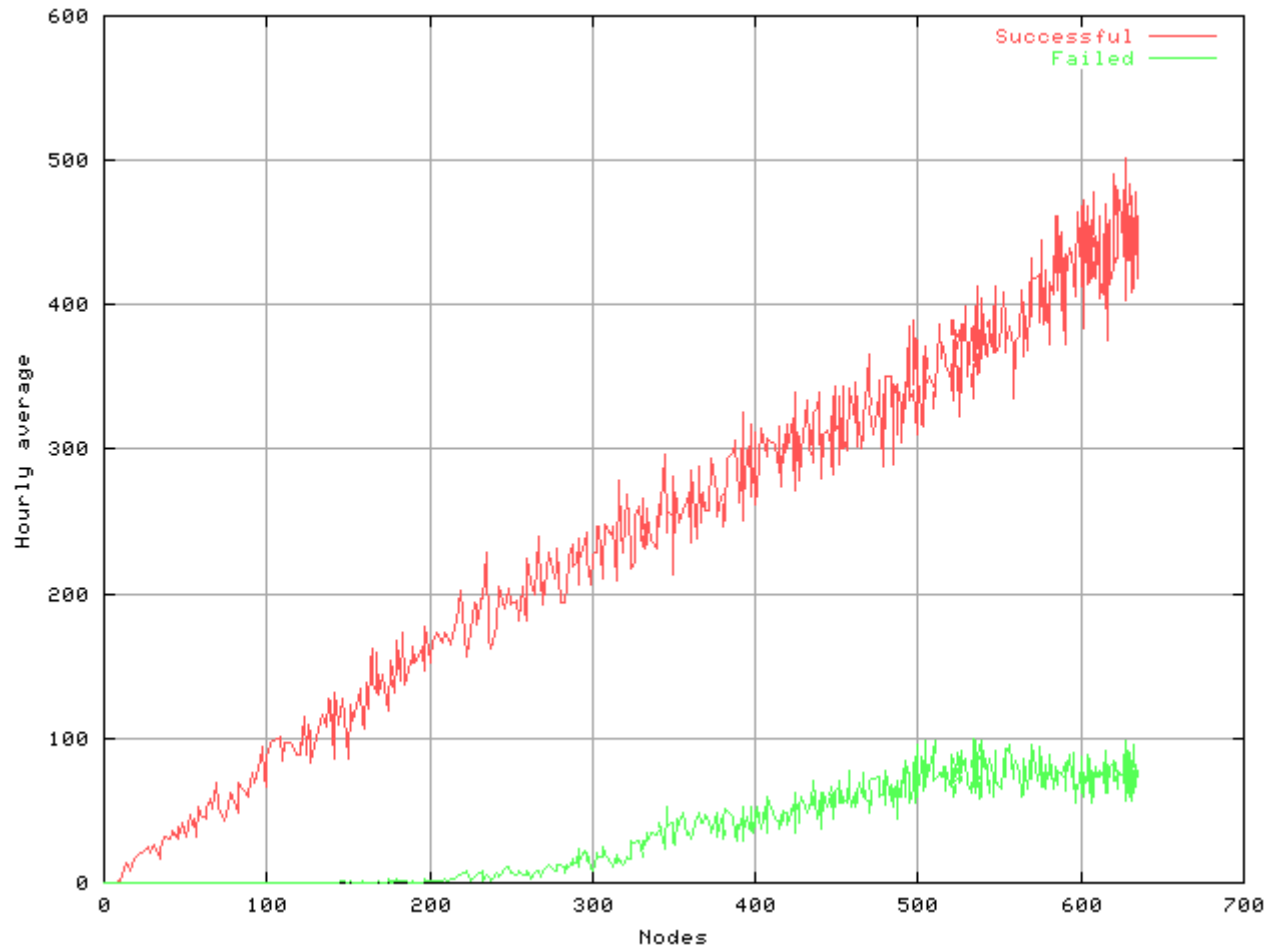
-- *Hubert Kirrman*

Freenet Simulations

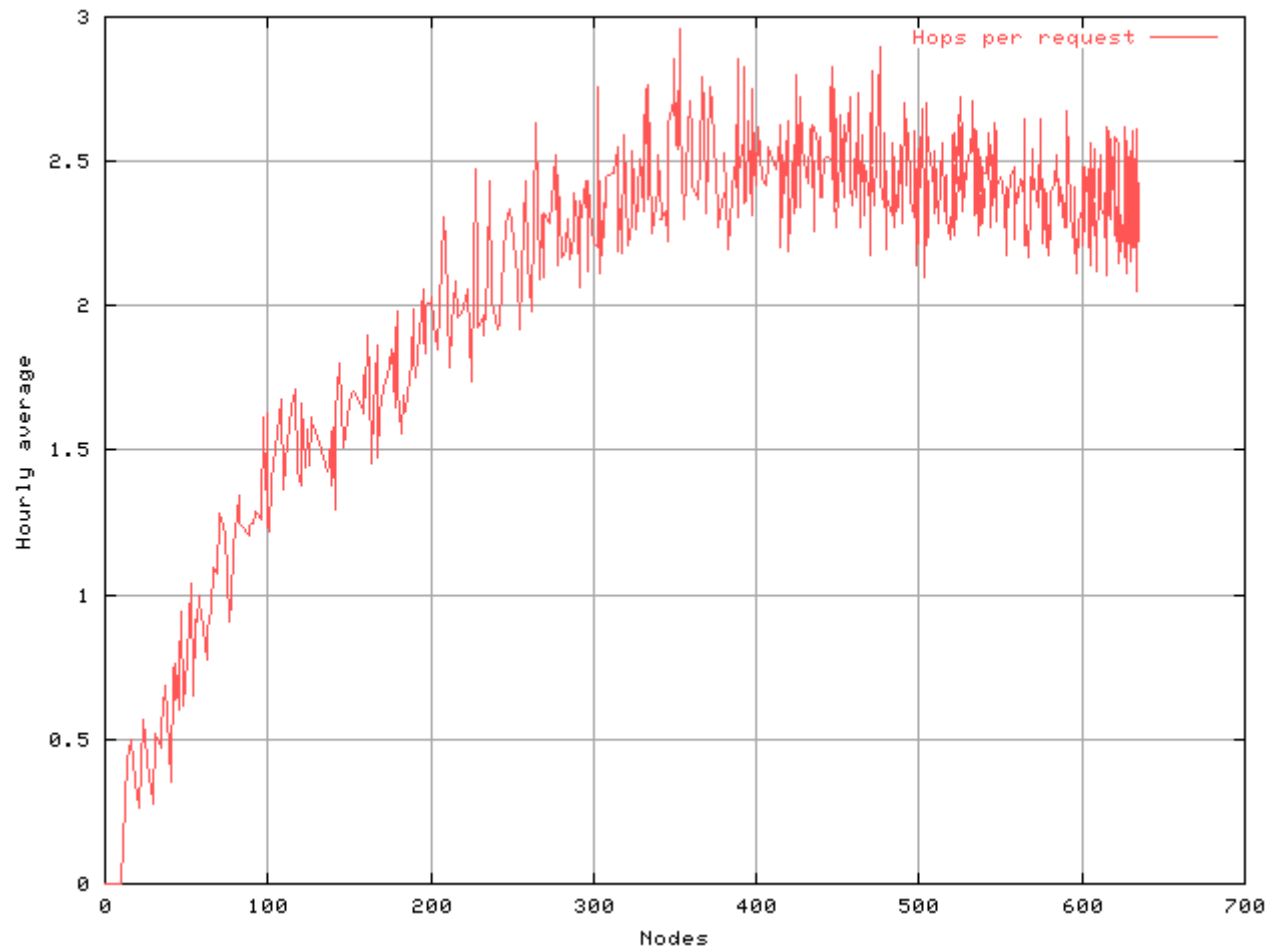
- Typical running time of 30 days network time.
- Growing to 800-1700 nodes.
- 200-400 000 simulated requests on.
- 10-20 000 documents.

One can see the Freenet topology emerging as the network runs.

Typical Freenet success rates



Typical Freenet query lengths



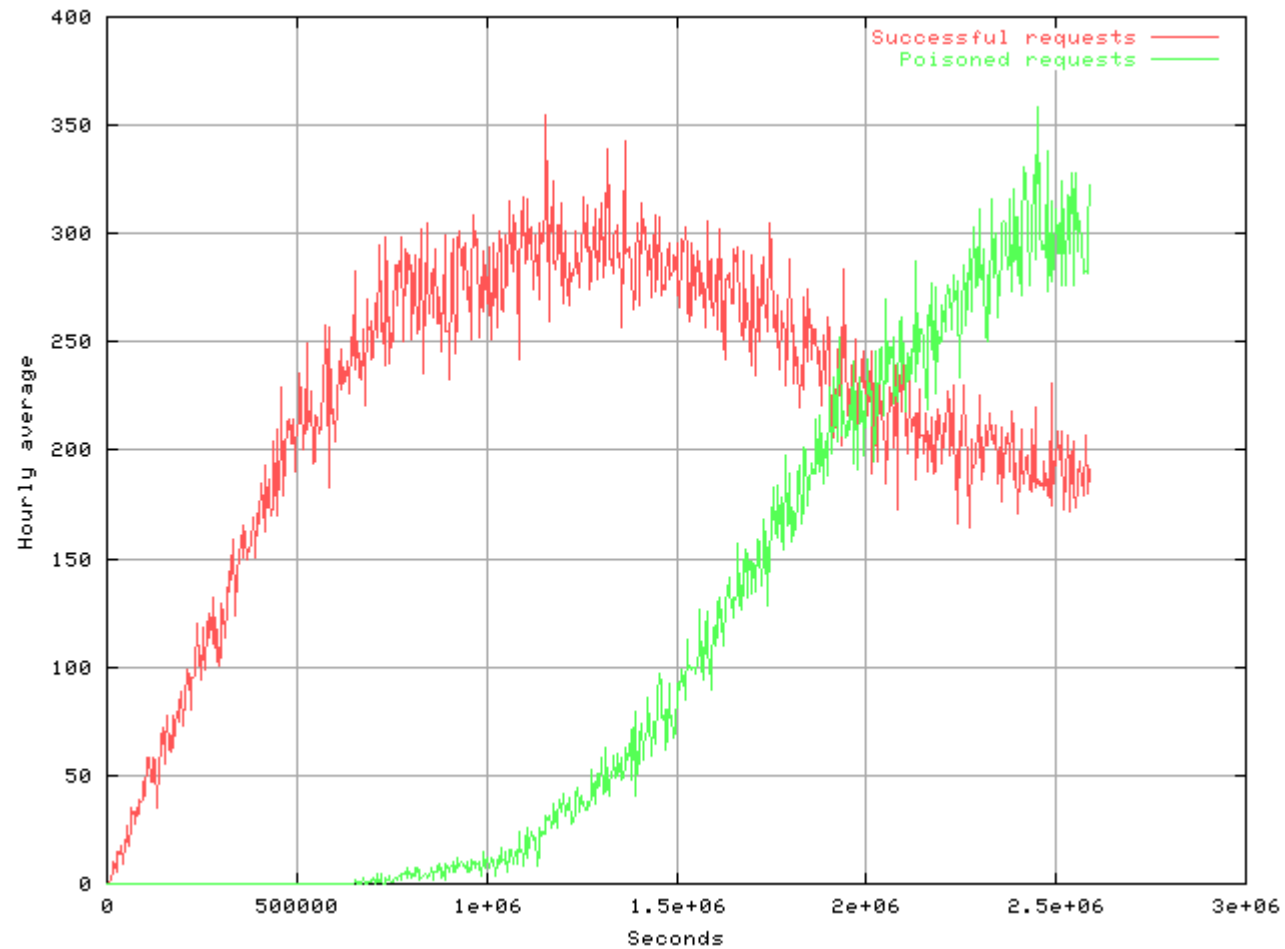
Freenet Cancer Attack

Freenet uses positive feedback to strengthen links to nodes that successfully provide data.

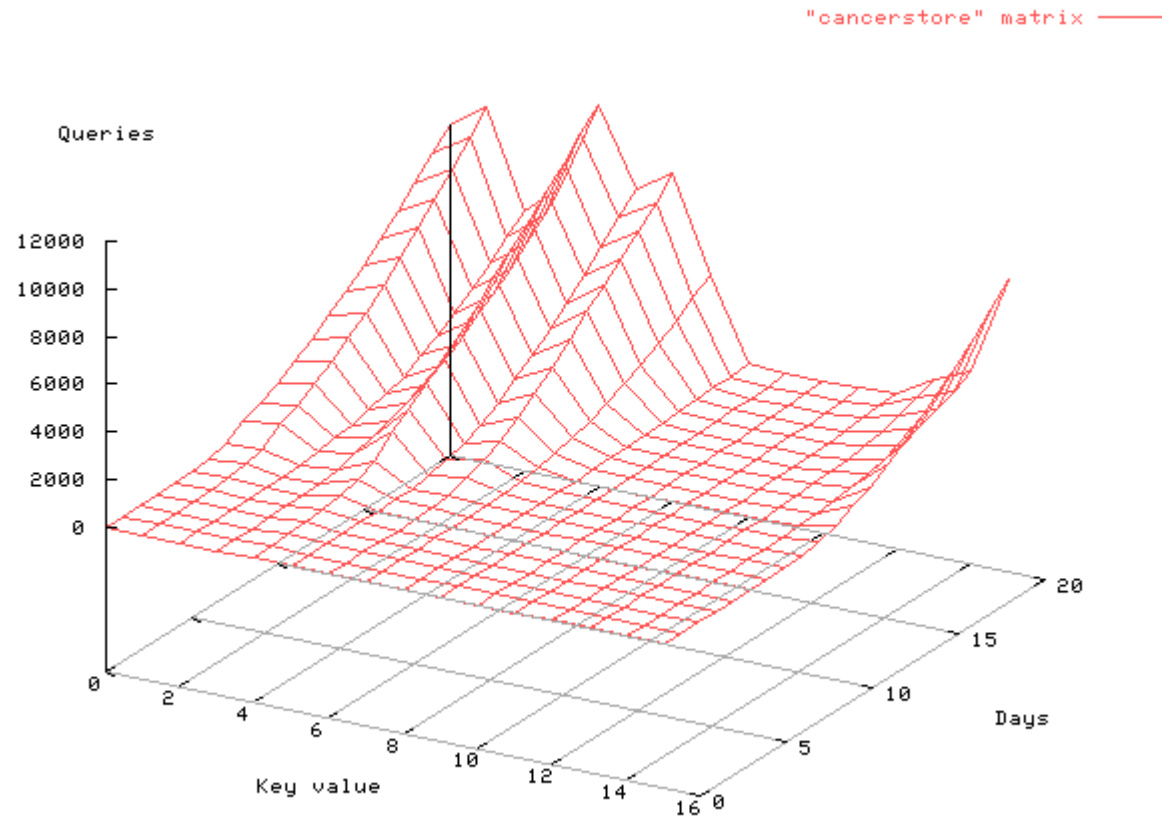
A node that always returns bogus data could create runaway positive feedback.

Somewhat like a cancer or a black hole.

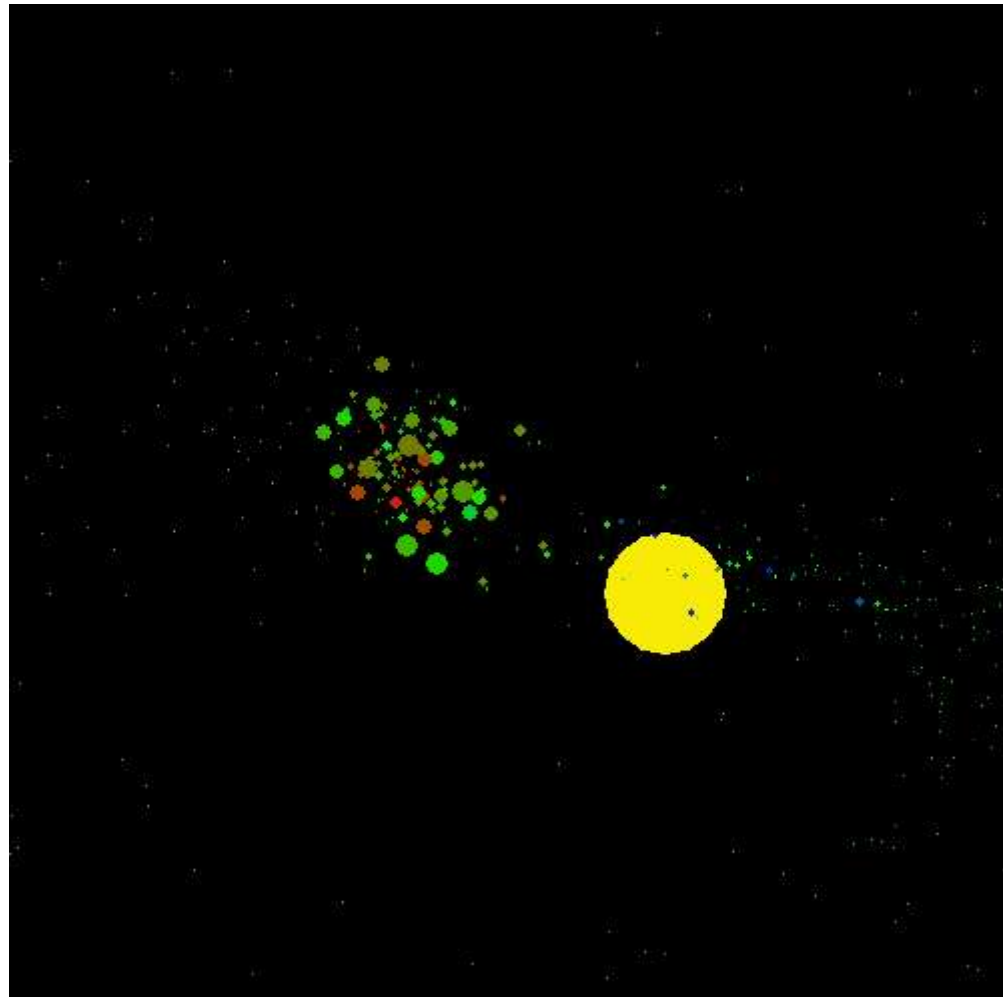
Freenet cancer results



Cancerous node request spread



Freenet cancer map



Cancer countermeasures

- The authenticity of all data is cryptographically verified at every step.
- If data cannot be verified, some sort of user validation is necessary.
- Honest cancers are still possible, but probably containable.

Freenet data removal attack

It is possible to target an attack at a certain piece of data if the data can be attacked without attacking the whole network.

Freenet request results provide a "DataSource", the address of node higher up the search path so lower nodes can update their routing tables.

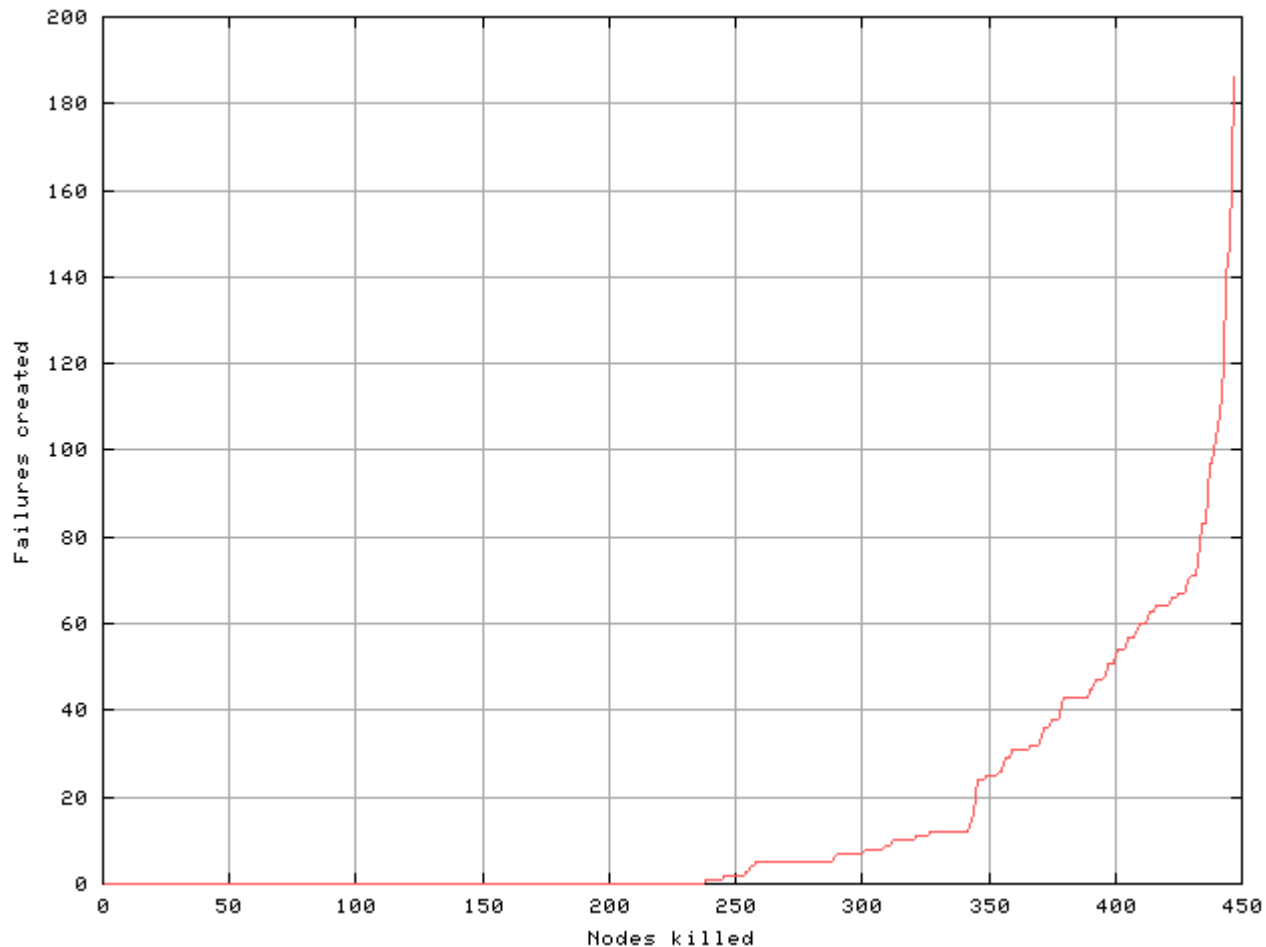
Freenet data removal attack methodology

Trying to remove a piece of data from Freenet:

- Request data.
- Read "DataSource" and eliminate that node using external method.
- Repeat ad infinitum.

Freenet data removal attack

There were about 500 nodes in this network when the attack started:



Data removal attack conclusions

- It took eliminating more than half the nodes in the network before any effect was seen.
- Other data actually fared worse than that targeted.
- Trying to target an attack in this manner is not effective on Freenet routing networks.

Other attacks on Freenet

- Flooding data into network.
- Request flood to over-cache data.
- Normal DOS attack using message flood.

Talk Gnutella

BLABLABLABLA

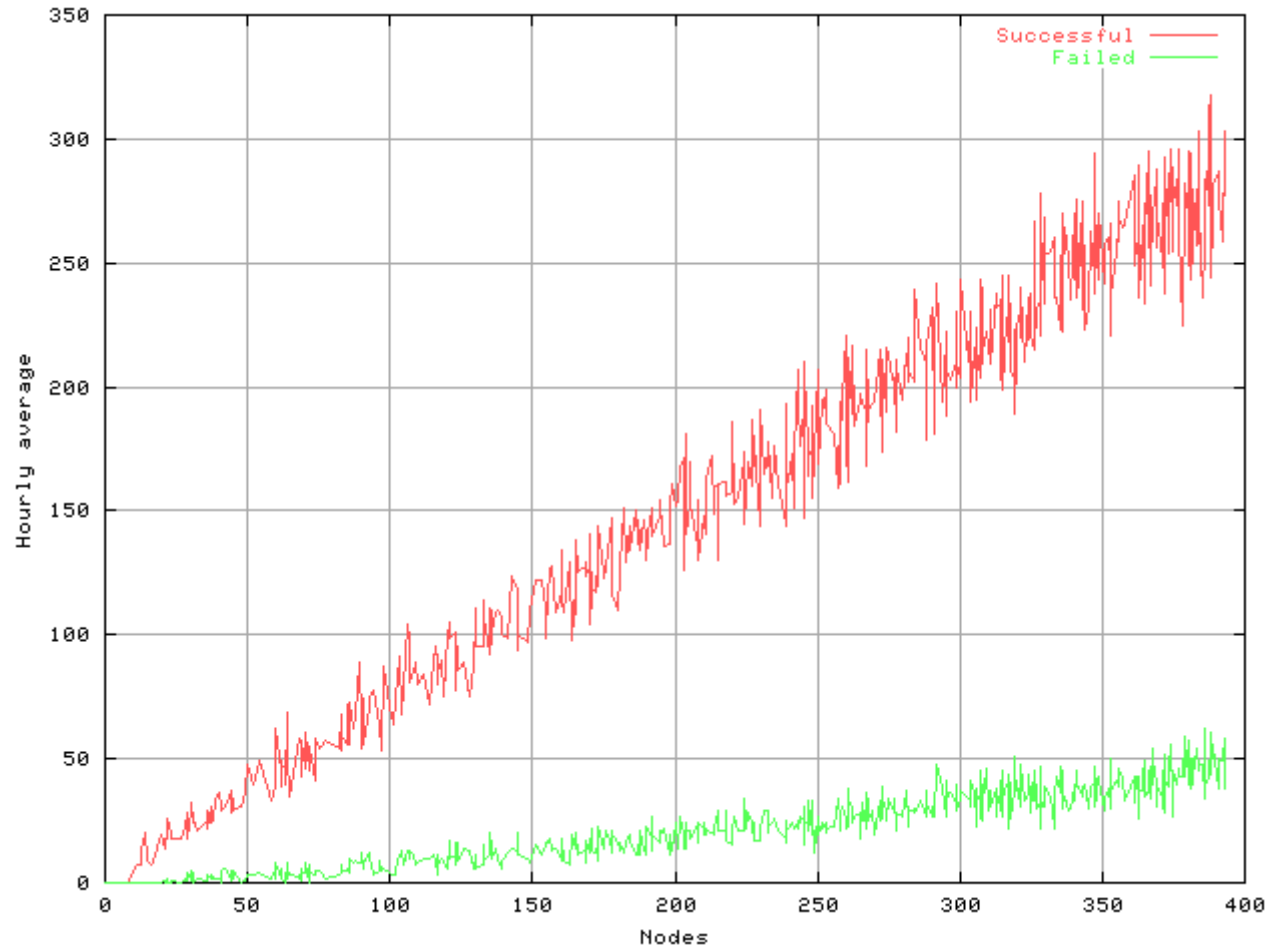
Hypermesh Simulations

The version implemented reminds most of *Tapestry* as used in the Oceanstore system.

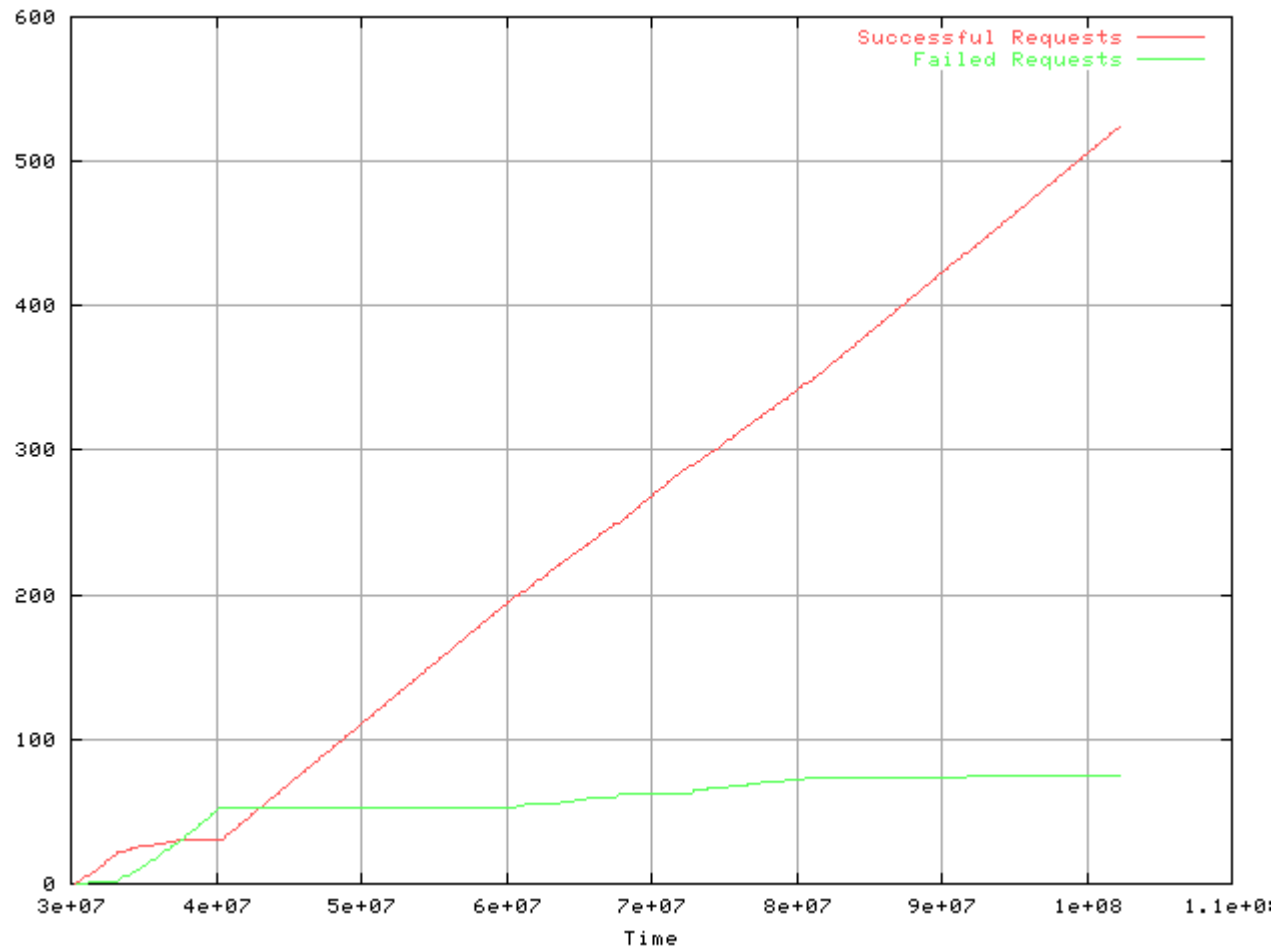
It is not completely equivalent however.

In theory, these networks should give 100% success, but on shifting networks it falls slightly short.

Hypermesh simulation success rates



Typical failure case



Data usurping nodes

Hypermesh nodes route by matching the Nodes ID value with the data's.

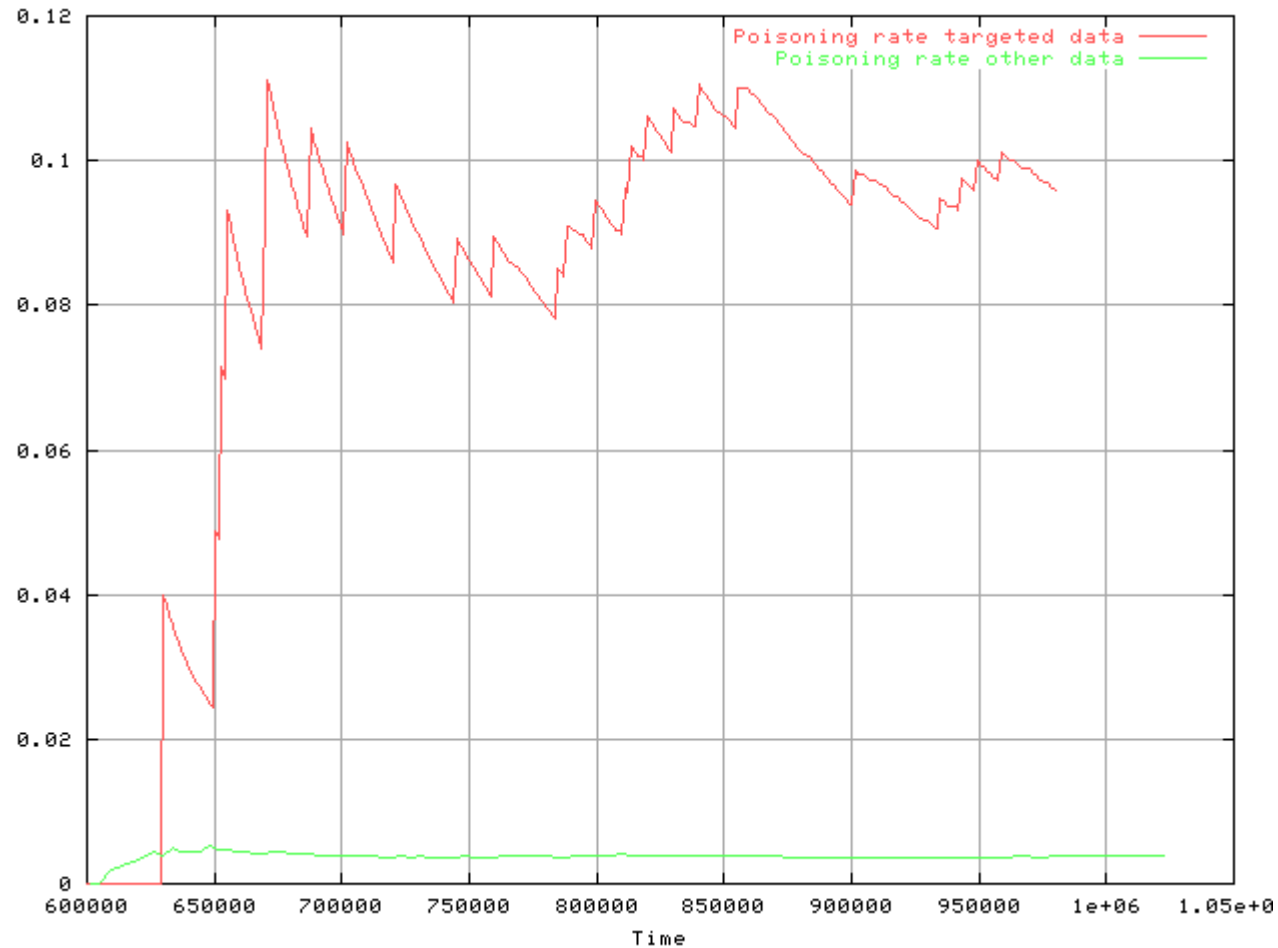
In a decentralized network, there is no way to control what ID a Node chooses.

A malicious node could choose it's ID to match data it wants to target.

Data usurping methodology

- Choose Node ID to match target data in as many coordinates as possible.
- Announce Node to network.
- Reject or poison all requests.

Data usurping results



Other attacks on Hypermesh networks

- Data removal (as per procedure against Freenet) should always work in limited number of steps.
- Generally the rigid nature makes it easy to target data.
- As always, flooding attacks are possible.

Attacks on other networks

- Global index topologies are sitting ducks to all sort of attacks (cf *Napster*).
- "Super node" type networks, such as *FastTrack* are susceptible to a number of attacks. Super nodes have a lot of power, and are not carefully chosen. Attempting to secure these using closed protocols hidden in user hostile software is short sighted, useless, and probably ultimately destructive.

Conclusion

- The attack vectors on P2P file publishing networks are almost endless.
- Being able to simulate the actual attacks is illustrative and helpful (and fun!)
- The Freenet Project will continue trying to find solutions.