Routing in The Dark

Scalable Searches in Dark P2P Networks

Ian Clarke and Oskar Sandberg

The Freenet Project

Introduction

- We have long been interested in decentralised “Peer to Peer” networks. Especially Freenet.

Introduction

- We have long been interested in decentralised "Peer to Peer" networks. Especially Freenet.
- But when individual users come under attack, decentralisation is not enough.

Introduction

- We have long been interested in decentralised "Peer to Peer" networks. Especially Freenet.
- But when individual users come under attack, decentralisation is not enough.
- Future networks may need to limit connections to trusted friends.

Introduction

- We have long been interested in decentralised "Peer to Peer" networks. Especially Freenet.
- But when individual users come under attack, decentralisation is not enough.
- Future networks may need to limit connections to trusted friends.
- The big question is: Can such networks be useful?

Overview of “Peer to Peer” networks

- Information is spread across many interconnected computers

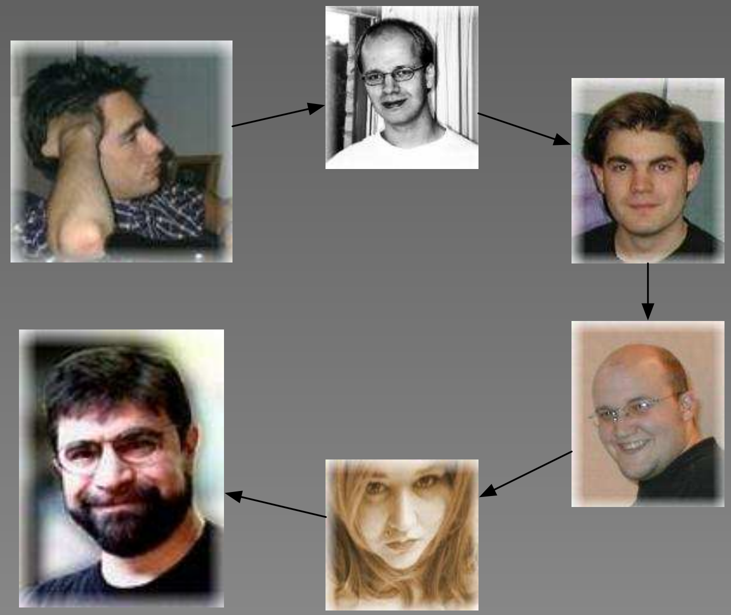
Overview of “Peer to Peer” networks

- Information is spread across many interconnected computers
- Users want to find information

Overview of “Peer to Peer” networks

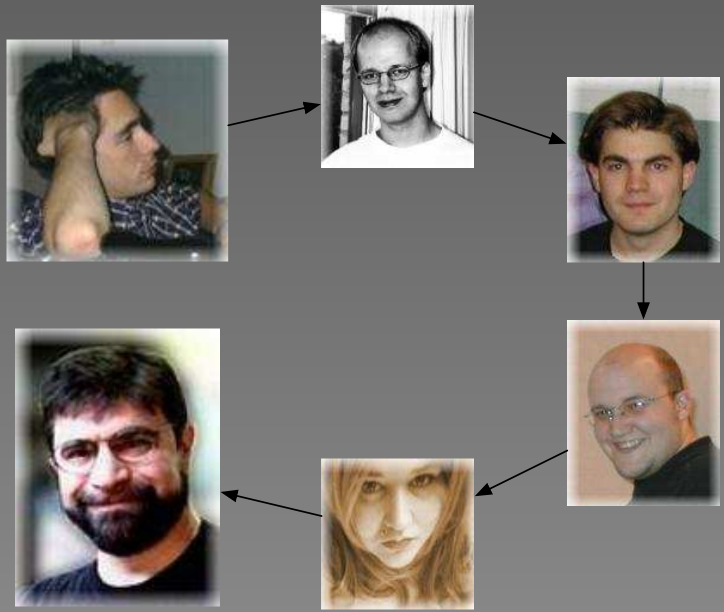
- Information is spread across many interconnected computers
- Users want to find information
- Some are centralised (eg. Napster), some are semi- centralised (eg. Kazaa), others are distributed (eg. Freenet)

The Small World Phenomenon



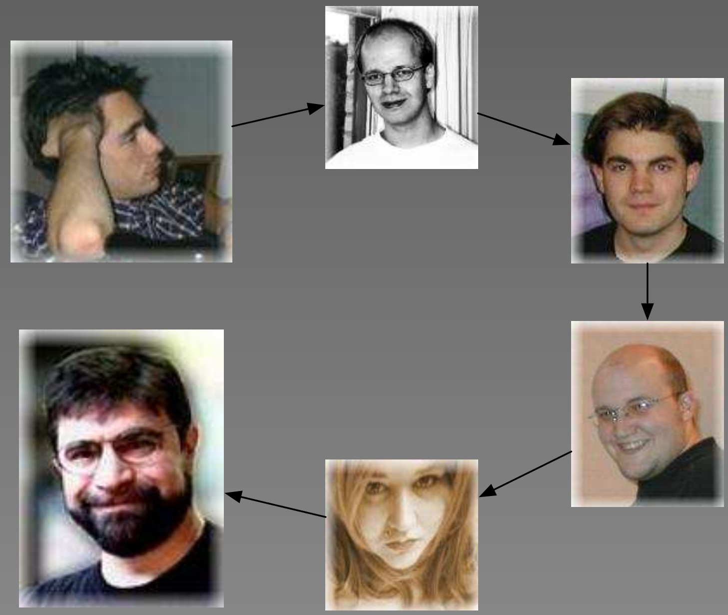
- In "Small world" networks short paths exist between any two peers

The Small World Phenomenon



- In "Small world" networks short paths exist between any two peers
- People tend to form this type of network (as shown by Milgram experiment)

The Small World Phenomenon



- In "Small world" networks short paths exist between any two peers
- People tend to form this type of network (as shown by Milgram experiment)
- Short paths may exist but they may not be easy to find

Navigable Small World Networks

- Concept of similarity or “closeness” between peers

Navigable Small World Networks

- Concept of similarity or “closeness” between peers
- Similar peers are more likely to be connected than dissimilar peers

Navigable Small World Networks

- Concept of similarity or “closeness” between peers
- Similar peers are more likely to be connected than dissimilar peers
- You can get from any one peer to any other simply by routing to the closest peer at each step

Navigable Small World Networks

- Concept of similarity or “closeness” between peers
- Similar peers are more likely to be connected than dissimilar peers
- You can get from any one peer to any other simply by routing to the closest peer at each step
- This is called “Greedy Routing”

Navigable Small World Networks

- Concept of similarity or “closeness” between peers
- Similar peers are more likely to be connected than dissimilar peers
- You can get from any one peer to any other simply by routing to the closest peer at each step
- This is called “Greedy Routing”
- Freenet and “Distributed Hash Tables” rely on this principal to find data in a scalable decentralised manner

Light P2P Networks

- Examples: Gnutella, Freenet, Distributed Hash Tables

Light P2P Networks

- Examples: Gnutella, Freenet, Distributed Hash Tables
- Advantage: Globally scalable with the right routing algorithm

Light P2P Networks

- Examples: Gnutella, Freenet, Distributed Hash Tables
- Advantage: Globally scalable with the right routing algorithm
- Disadvantage: Vulnerable to “harvesting”, ie. people you don’t know can easily discover whether you are part of the network

Dark or “Friend to Friend” P2P Networks

- Peers only communicate directly with “trusted” peers

Dark or “Friend to Friend” P2P Networks

- Peers only communicate directly with “trusted” peers
- Examples: Waste

Dark or “Friend to Friend” P2P Networks

- Peers only communicate directly with “trusted” peers
- Examples: Waste
- Advantage: Only your trusted friends know you are part of the network

Application

How can we apply small world theory to routing in a Dark peer to peer network?

Application

How can we apply small world theory to routing in a Dark peer to peer network?

- A Darknet is, essentially, a social network of peoples trusted relationships.

Application

How can we apply small world theory to routing in a Dark peer to peer network?

- A Darknet is, essentially, a social network of peoples trusted relationships.
- If people can route in a social network, then it should be possible for computers.

Application

How can we apply small world theory to routing in a Dark peer to peer network?

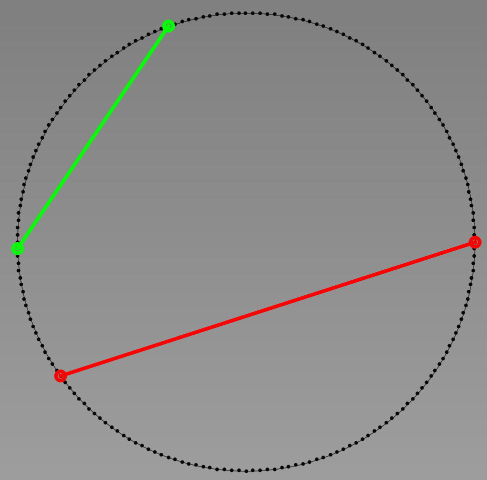
- A Darknet is, essentially, a social network of peoples trusted relationships.
- If people can route in a social network, then it should be possible for computers.
- Jon Kleinberg explained in 2000 how small world networks can be navigable.

Kleinberg's Result

- The possibility of routing efficiently depends on the proportion of connections that have different lengths with respect to the "position" of the nodes.

Kleinberg's Result

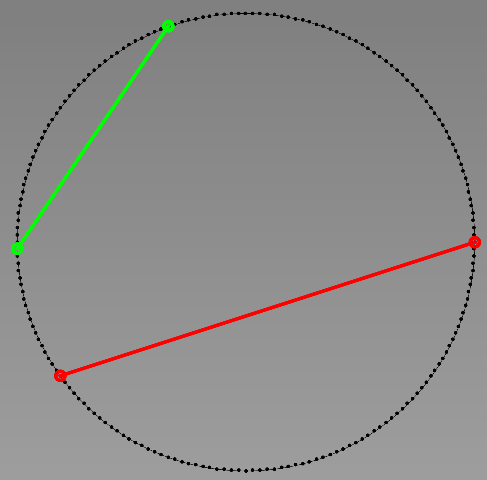
- The possibility of routing efficiently depends on the proportion of connections that have different lengths with respect to the "position" of the nodes.
- If the positions are in a ring, the proportion of connections with a certain length should be inverse to the length:



Kleinberg's Result

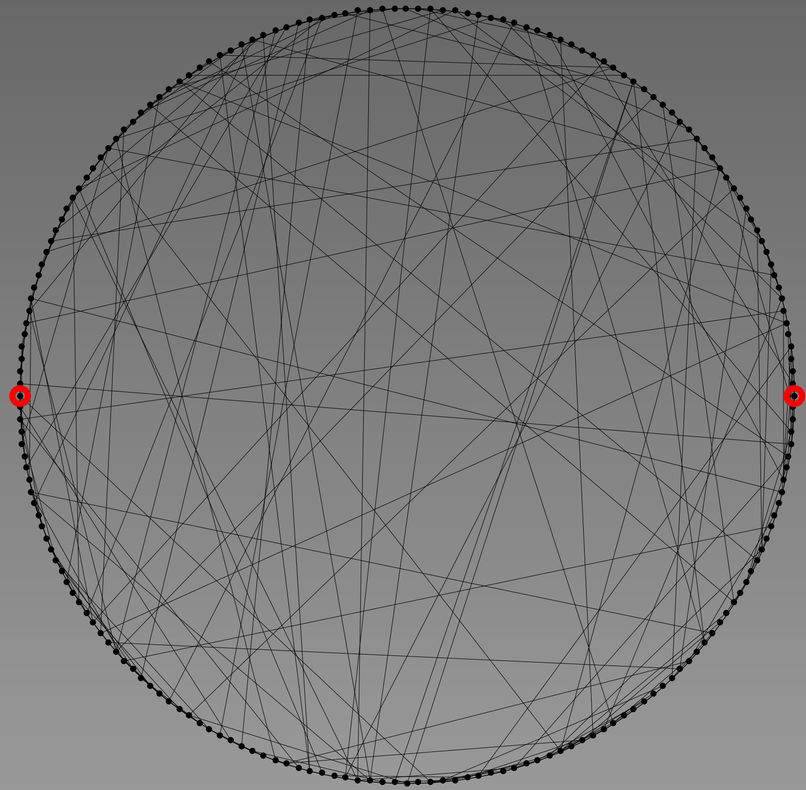
- The possibility of routing efficiently depends on the proportion of connections that have different lengths with respect to the "position" of the nodes.

- If the positions are in a ring, the proportion of connections with a certain length should be inverse to the length:

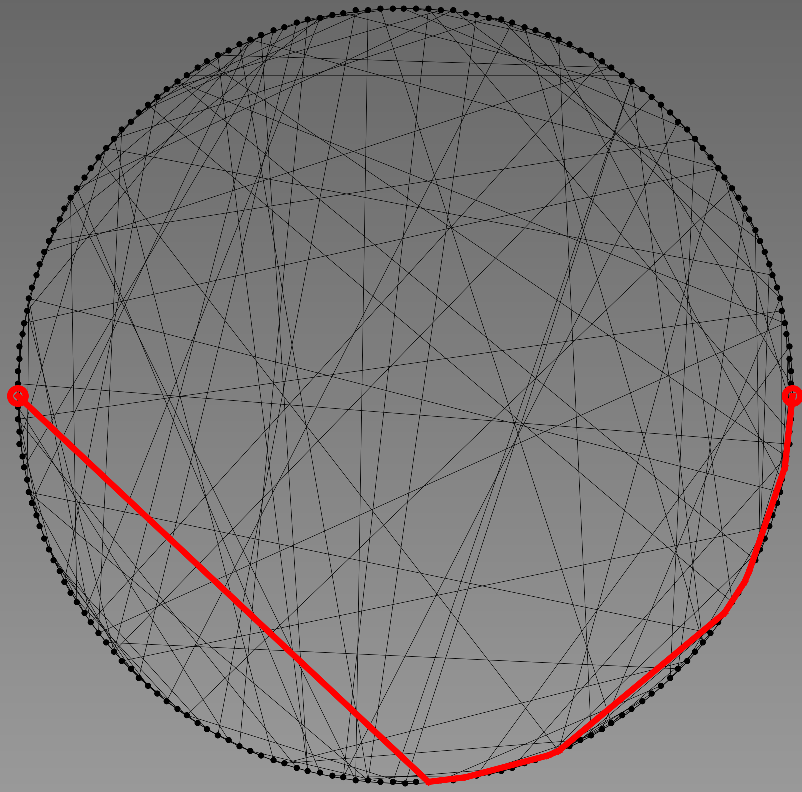


- In this case a simple *greedy routing* algorithm performs in $O(\log^2 n)$ steps.

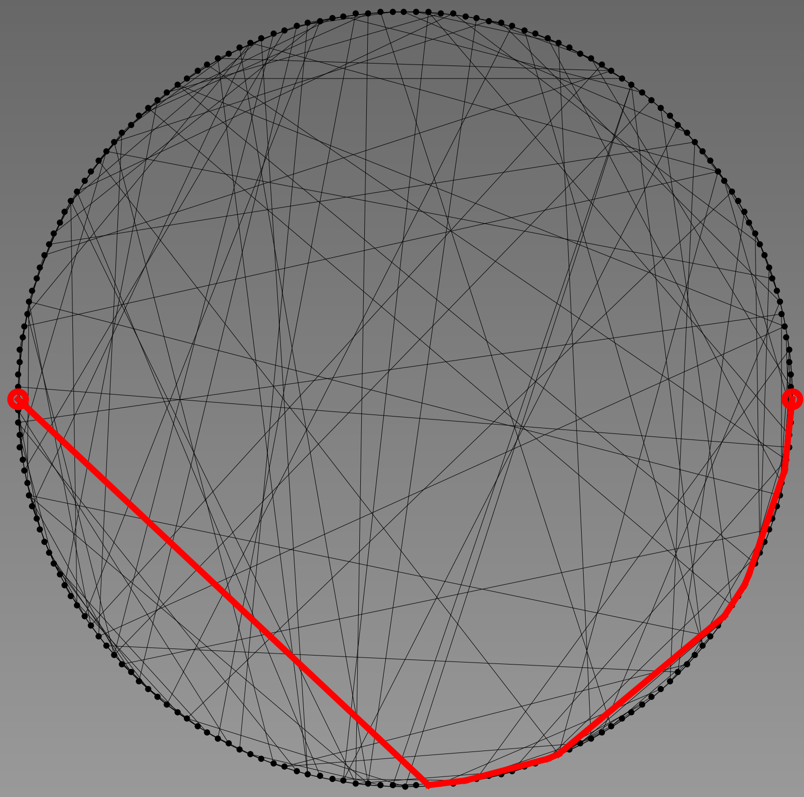
Kleinbergs Result, cont.



Kleinbergs Result, cont.



Kleinbergs Result, cont.



But in a social network, how do we see if one person is closer to the destination than another?

Application, cont.

Is Alice closer to Harry than Bob?

Application, cont.

Is Alice closer to Harry than Bob?

- In real life, people presumably use a large number of factors to decide this. Where do they live? What are their jobs? What are their interests?

Application, cont.

Is Alice closer to Harry than Bob?

- In real life, people presumably use a large number of factors to decide this. Where do they live? What are their jobs? What are their interests?
- One cannot, in practice, expect a computer to route based on such things.

Application, cont.

Is Alice closer to Harry than Bob?

- In real life, people presumably use a large number of factors to decide this. Where do they live? What are their jobs? What are their interests?
- One cannot, in practice, expect a computer to route based on such things.
- Instead, we let the network tell us!

Application, cont.

- Kleinberg's model suggests: there should be few long connections, and many short ones.

Application, cont.

- Kleinberg's model suggests: there should be few long connections, and many short ones.
- We can assign numerical identities placing nodes in a circle, and do it in such a way that this is fulfilled.

Application, cont.

- Kleinberg's model suggests: there should be few long connections, and many short ones.
- We can assign numerical identities placing nodes in a circle, and do it in such a way that this is fulfilled.
- Then greedy route with respect to these numerical identities.

The Method

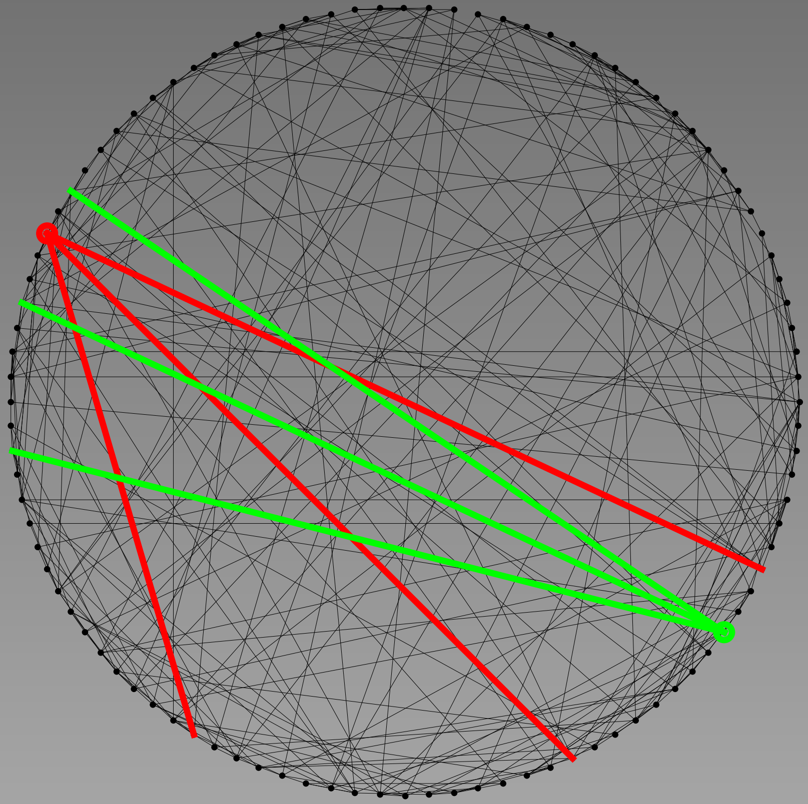
- When nodes join the network, they choose a position on the circle randomly.

The Method

- When nodes join the network, they choose a position on the circle randomly.
- They then switch positions with other nodes, so as to minimize the product of the edge distances.

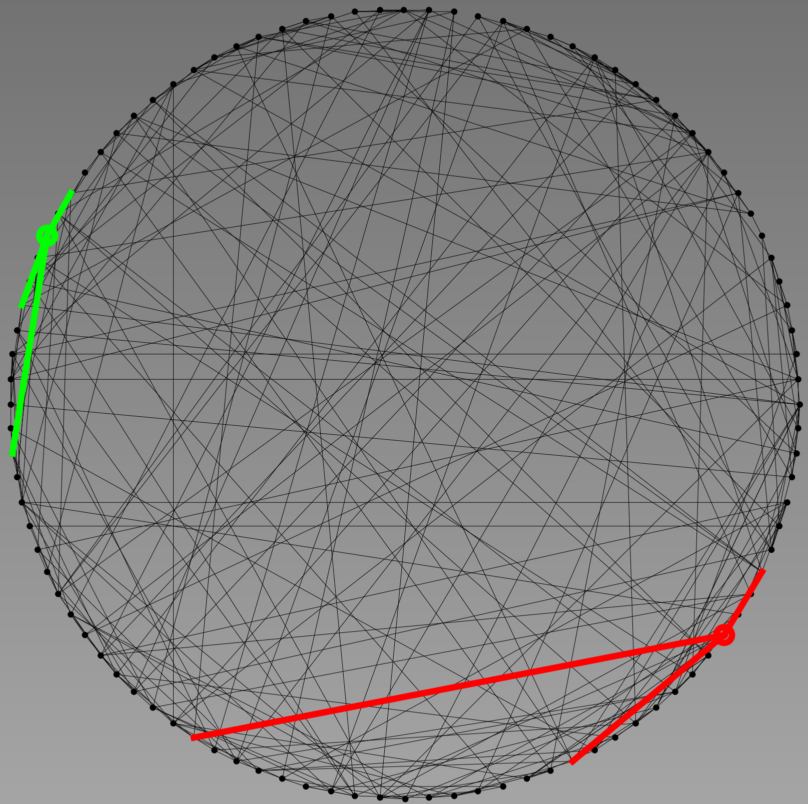
The Method, cont.

An advantageous switch of position:



The Method, cont.

An advantageous switch of position:



The Method, cont.

Some notes:

The Method, cont.

Some notes:

- Switching is essential!

The Method, cont.

Some notes:

- Switching is essential!
- Because this is an ongoing process as the network grows (and shrinks) it will be difficult to keep permanent positions.

The Algorithm

- Two nodes are chosen in some random fashion, and attempt to switch.

The Algorithm

- Two nodes are chosen in some random fashion, and attempt to switch.
- They calculate ℓ_b as the product of all the lengths of their current connections. Then they calculate ℓ_a as the product of what all their respective connection lengths would be after they switched.

The Algorithm

- Two nodes are chosen in some random fashion, and attempt to switch.
- They calculate ℓ_b as the product of all the lengths of their current connections. Then they calculate ℓ_a as the product of what all their respective connection lengths would be after they switched.
- If $\ell_b > \ell_a$ they switch. Otherwise they switch with probability ℓ_b / ℓ_a.

The Algorithm, cont.

Let $d(z)$ give the degree (number of connections) of a node z, and let $e_i(z)$ and $e_i'(z)$ be distance of z's i- th connection before and after a switch occurs. Let nodes x and y be the ones attempting to switch. Calculate:

$$p = \frac{\ell(a)}{\ell(b)} = \frac{\prod_{i=1}^{d(x)} e_i(x) \prod_{i=1}^{d(y)} e_i(y)}{\prod_{i=1}^{d(x)} e_i'(x) \prod_{i=1}^{d(y)} e_i'(y)}$$

x and y will complete the switch with probability $\min(1, p)$. Otherwise we leave the network as it is.

The Algorithm, cont.

- This is an application of the Metropolis- Hastings algorithm.

The Algorithm, cont.

- This is an application of the Metropolis- Hastings algorithm.
- Because there is a greater chance of moving to positions with shorter connection distances, it will tend to minimize the product of the distances.

The Algorithm, cont.

- This is an application of the Metropolis- Hastings algorithm.
- Because there is a greater chance of moving to positions with shorter connection distances, it will tend to minimize the product of the distances.
- Because the probability of making a switch is never zero, it cannot get stuck in a bad configuration (a local minima).

The Algorithm, cont.

- How do nodes choose each other to attempt to switch?

The Algorithm, cont.

- How do nodes choose each other to attempt to switch?
- Any method will work in theory, but some will work better than others. Only switching with neighbors does not seem to work in practice.

The Algorithm, cont.

- How do nodes choose each other to attempt to switch?
- Any method will work in theory, but some will work better than others. Only switching with neighbors does not seem to work in practice.
- Our current method is to do a short random walk starting at one of the nodes and terminating at the other.

Simulations

We have simulated networks in three different modes:

Simulations

We have simulated networks in three different modes:

- Random walk search: “random”.

Simulations

We have simulated networks in three different modes:

- Random walk search: “random”.
- Greedy routing in Kleinberg’s model with identities as when it was constructed: “good”.

Simulations

We have simulated networks in three different modes:

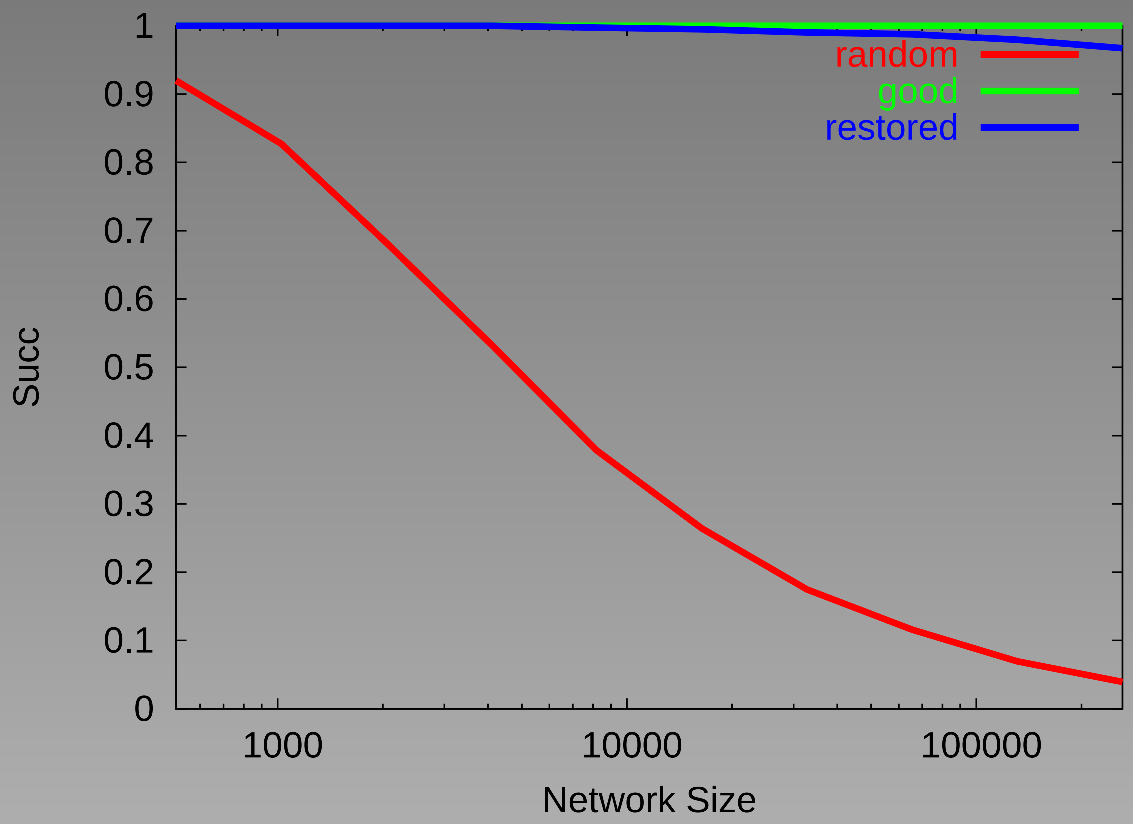
- Random walk search: “random”.
- Greedy routing in Kleinberg’s model with identities as when it was constructed: “good”.
- Greedy routing in Kleinberg’s model with identities assigned according to our algorithm (2000 iterations per node): “restored”.

Simulations, cont.

The proportion of queries that succeeded within $(\log_2 n)^2$ steps, where n is the network size:

Simulations, cont.

The proportion of queries that succeeded within $(\log_2 n)^2$ steps, where n is the network size:

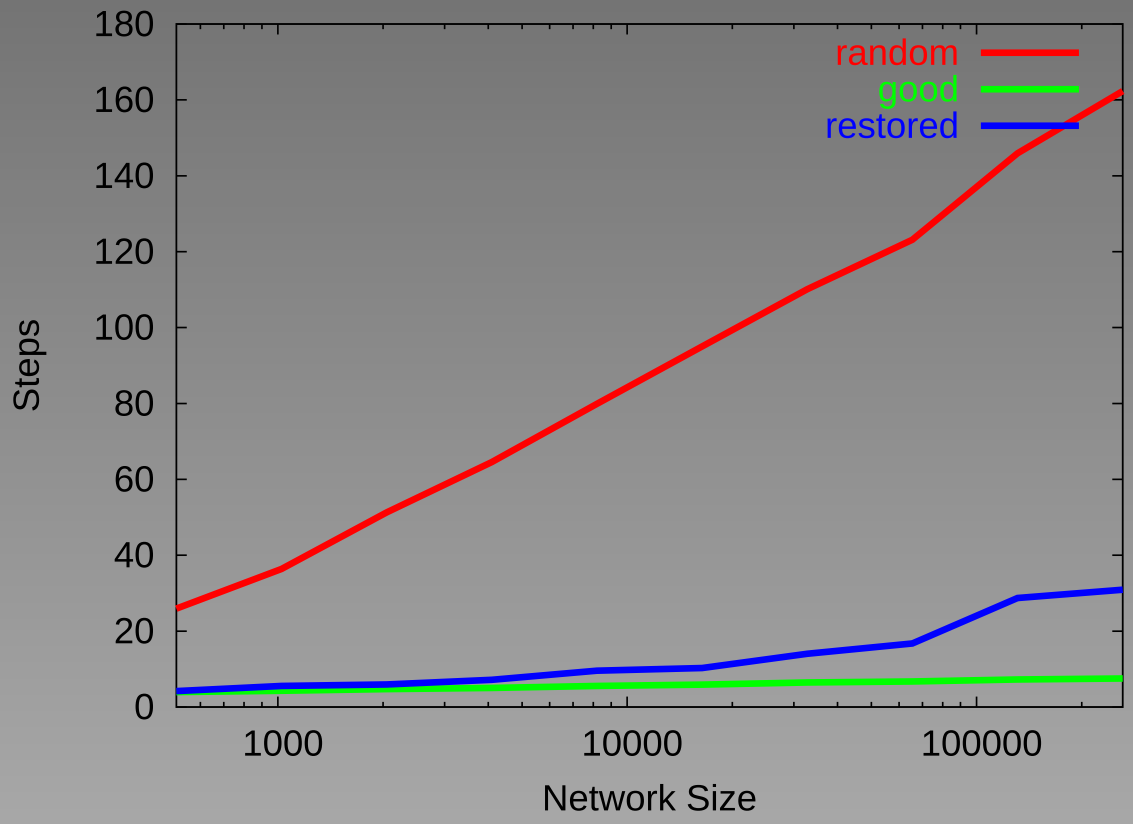


Simulations, cont.

The average length of the successful routes:

Simulations, cont.

The average length of the successful routes:



Results

- Simulated networks are only so interesting, what about the real world?

Results

- Simulated networks are only so interesting, what about the real world?
- We borrowed some data from orkut.com. 2196 people were spidered, starting with Ian.

Results, cont.

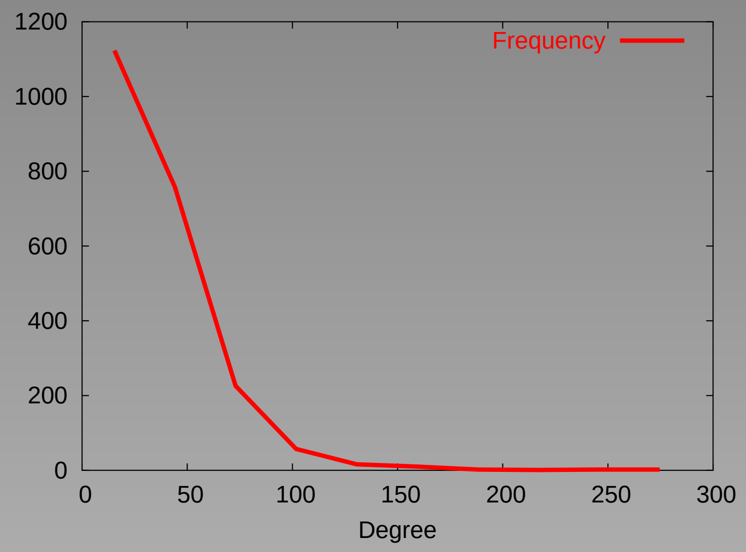
- The set was spidered so as to be comparatively dense (average 36.7 connections per person).

Results, cont.

- The set was spidered so as to be comparatively dense (average 36.7 connections per person).
- It contains mostly American techies and programmers. Some are probably in this room. (No Brazilians...)

Results, cont.

- The set was spidered so as to be comparatively dense (average 36.7 connections per person).
- It contains mostly American techies and programmers. Some are probably in this room. (No Brazilians...)
- The degree distribution is approximately Power-Law:



Results, cont.

Searching the Orkut dataset, for a maximum of $\log_2(n)^2$ steps.

	Success Rate	Mean Steps
Random Search		
Our Algorithm		

Results, cont.

Searching the Orkut dataset, for a maximum of $\log_2(n)^2$ steps.

	Success Rate	Mean Steps
Random Search	0.72	43.85
Our Algorithm		

Results, cont.

Searching the Orkut dataset, for a maximum of $\log_2(n)^2$ steps.

	Success Rate	Mean Steps
Random Search	0.72	43.85
Our Algorithm	0.97	7.714

Results

Clipping degree at 40 connections. (24.2 connections per person.)

	Success Rate	Mean Steps
Random Search		
Our Algorithm		

Results

Clipping degree at 40 connections. (24.2 connections per person.)

	Success Rate	Mean Steps
Random Search	0.51	50.93
Our Algorithm		

Results

Clipping degree at 40 connections. (24.2 connections per person.)

	Success Rate	Mean Steps
Random Search	0.51	50.93
Our Algorithm	0.98	10.90

Results

Clipping degree at 40 connections. (24.2 connections per person.)

	Success Rate	Mean Steps
Random Search	0.51	50.93
Our Algorithm	0.98	10.90

Our algorithm takes advantage of there being people who have many connections, but it does not depend on them.

Practical Concerns

- So the theory works, but how does one implement such a network in practice?

Practical Concerns

- So the theory works, but how does one implement such a network in practice?
- Key concerns:

Practical Concerns

- So the theory works, but how does one implement such a network in practice?
- Key concerns:
  - Preventing malicious behaviour

Practical Concerns

- So the theory works, but how does one implement such a network in practice?
- Key concerns:
  - Preventing malicious behaviour
  - Ensuring ease of use

Practical Concerns

- So the theory works, but how does one implement such a network in practice?
- Key concerns:
  - Preventing malicious behaviour
  - Ensuring ease of use
  - Storing data

Preventing Malicious Behaviour

Threats:

- Selection of identity to attract certain data

Preventing Malicious Behaviour

Threats:

- Selection of identity to attract certain data
- Manipulation of other node’s identities

Ensuring ease of use

- Peers will need to be “always on”

Ensuring ease of use

- Peers will need to be “always on”
- Peer introduction

Ensuring ease of use

- Peers will need to be “always on”
- Peer introduction
  - Email

Ensuring ease of use

- Peers will need to be “always on”
- Peer introduction
  - Email
  - Phone

Ensuring ease of use

- Peers will need to be “always on”
- Peer introduction
  - Email
  - Phone
  - Trusted third party

Ensuring ease of use

- Peers will need to be “always on”
- Peer introduction
  - Email
  - Phone
  - Trusted third party
- What about NATs and firewalls

Ensuring ease of use

- Peers will need to be “always on”
- Peer introduction
  - Email
  - Phone
  - Trusted third party
- What about NATs and firewalls
  - Could use UDP hole- punching (as used by Dijjer, Skype)

Ensuring ease of use

- Peers will need to be “always on”
- Peer introduction
  - Email
  - Phone
  - Trusted third party
- What about NATs and firewalls
  - Could use UDP hole- punching (as used by Dijjer, Skype)
  - Would require third- party for negotiation

Storing Data

- We can store data as in a caching Distributed Hash Table (similar to Freenet)

Storing Data

- We can store data as in a caching Distributed Hash Table (similar to Freenet)
- We can also route directly between two peers if we know their identities

Storing Data

- We can store data as in a caching Distributed Hash Table (similar to Freenet)
- We can also route directly between two peers if we know their identities
  - Problem: Identities change

Storing Data

- We can store data as in a caching Distributed Hash Table (similar to Freenet)
- We can also route directly between two peers if we know their identities
  - Problem: Identities change
- We could employ a "crossing paths" approach

Storing Data

- We can store data as in a caching Distributed Hash Table (similar to Freenet)
- We can also route directly between two peers if we know their identities
  - Problem: Identities change
- We could employ a "crossing paths" approach
  - Both peers route towards the same random identity

Storing Data

- We can store data as in a caching Distributed Hash Table (similar to Freenet)
- We can also route directly between two peers if we know their identities
  - Problem: Identities change
- We could employ a "crossing paths" approach
  - Both peers route towards the same random identity
  - When paths cross a connection is established

Conclusion

We believe very strongly that building a navigable, scalable Darknet is possible. *And we intend to do it!*

Conclusion

We believe very strongly that building a navigable, scalable Darknet is possible. *And we intend to do it!*

- There is still much work to do on the theory.

Conclusion

We believe very strongly that building a navigable, scalable Darknet is possible. *And we intend to do it!*

- There is still much work to do on the theory.
    - Can other models work better?

Conclusion

We believe very strongly that building a navigable, scalable Darknet is possible. *And we intend to do it!*

- There is still much work to do on the theory.
  - Can other models work better?
  - Can we find better selection functions for switching?

Conclusion

We believe very strongly that building a navigable, scalable Darknet is possible. *And we intend to do it!*

- There is still much work to do on the theory.
  - Can other models work better?
  - Can we find better selection functions for switching?
  - It needs to be tested on more data.

Conclusion, cont.

- We have learned the hard way that practice is more difficult than theory.

Conclusion, cont.

- We have learned the hard way that practice is more difficult than theory.
  - Security issues are very important.

Conclusion, cont.

- We have learned the hard way that practice is more difficult than theory.
  - Security issues are very important.
  - How the network is deployed will affect how well it works.

Conclusion, cont.

- We have learned the hard way that practice is more difficult than theory.
  - Security issues are very important.
  - How the network is deployed will affect how well it works.

People who are interested can join the discussion at *http://freenetproject.org/*.

Long Live the Darknet!