# Distributed Routing in Small Worlds - Darknet

Oskar Sandberg

# The Darknet

- Peer- to- peer (P2P) networks are *virtual* networks created when users' computers connect over the Internet.

# The Darknet

- Peer- to- peer (P2P) networks are *virtual* networks created when users' computers connect over the Internet.
- They are often used for file- sharing: users want to publish files for others in the network to see.

# The Darknet

- Peer- to- peer (P2P) networks are *virtual* networks created when users' computers connect over the Internet.
- They are often used for file- sharing: users want to publish files for others in the network to see.
- Traditionally, P2P networks have used centralized or collaborative indexing systems (DHTs) to find data.

# The Darknet

- Peer- to- peer (P2P) networks are *virtual* networks created when users' computers connect over the Internet.
- They are often used for file- sharing: users want to publish files for others in the network to see.
- Traditionally, P2P networks have used centralized or collaborative indexing systems (DHTs) to find data.
- Clients ask the index where a file is, and then ask the computer that contains the file for it.

# The Darknet, cont.

- There is no way of hiding in these networks. If you have a file: everybody knows.

# The Darknet, cont.

- There is no way of hiding in these networks. If you have a file: everybody knows.
- Networks, and users, are under attack from governments and others.

# The Darknet, cont.

- There is no way of hiding in these networks. If you have a file: everybody knows.
- Networks, and users, are under attack from governments and others.
- A “Darknet” is a P2P network where your computer talks only to your trusted friends. Nobody else knows that you are in the network at all.

# The Darknet, cont.

- There is no way of hiding in these networks. If you have a file: everybody knows.
- Networks, and users, are under attack from governments and others.
- A “Darknet” is a P2P network where your computer talks only to your trusted friends. Nobody else knows that you are in the network at all.
- To access data in such a network, we must find paths though the social graph.

# The Idea

How can we apply small-world theory to algorithmically route in a Darknet?

# The Idea

How can we apply small-world theory to algorithmically route in a Darknet?

- If people can route in a social network, then it should be possible for computers.

# The Idea

How can we apply small-world theory to algorithmically route in a Darknet?

- If people can route in a social network, then it should be possible for computers.
- Kleinberg's small-world model predicts that greedy routing is an efficient algorithm, if one exists.

# The Idea

How can we apply small-world theory to algorithmically route in a Darknet?

- If people can route in a social network, then it should be possible for computers.
- Kleinberg's small-world model predicts that greedy routing is an efficient algorithm, if one exists.
- But in a social network, how do we see if one person is closer to the destination than another?

# The Idea, cont.

Is Alice closer to Harry than Bob?

# The Idea, 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?

# The Idea, 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.

# The Idea, 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!

# The Idea, cont.

- Kleinberg's model tells us what proportion of the links should have different lengths for efficient routing to be possible.

# The Idea, cont.

- Kleinberg's model tells us what proportion of the links should have different lengths for efficient routing to be possible.
- We can assign numerical identities placing nodes in a circle in a manner that attempts to fulfill this property.

# The Idea, cont.

- Kleinberg's model tells us what proportion of the links should have different lengths for efficient routing to be possible.
- We can assign numerical identities placing nodes in a circle in a manner that attempts to fulfill this property.
- Then we greedy route with respect to these numerical identities.

# The Method

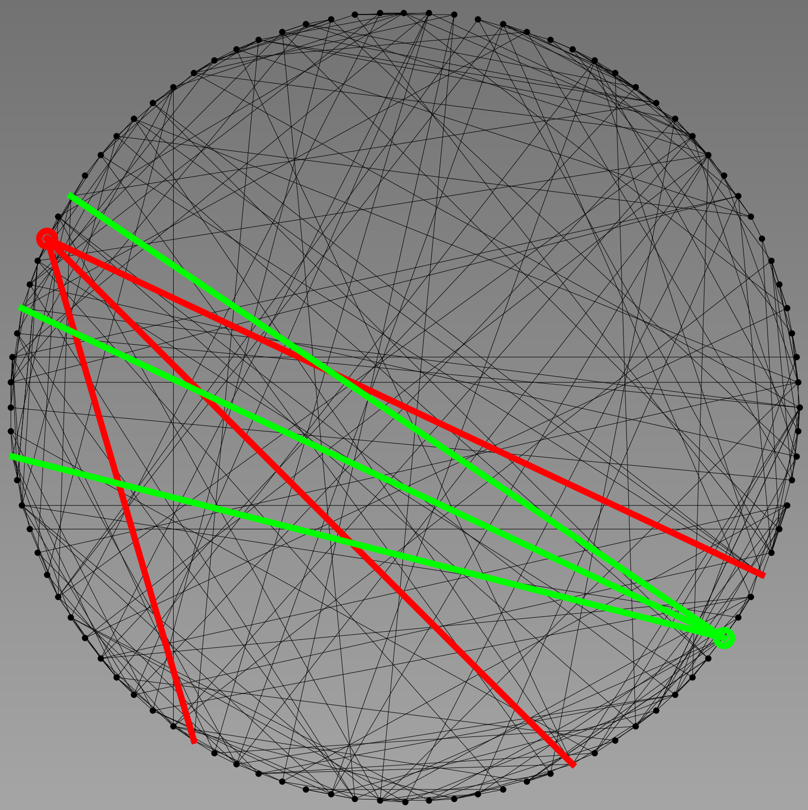
- When nodes join the network, they choose a position on the circle randomly.

# 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.

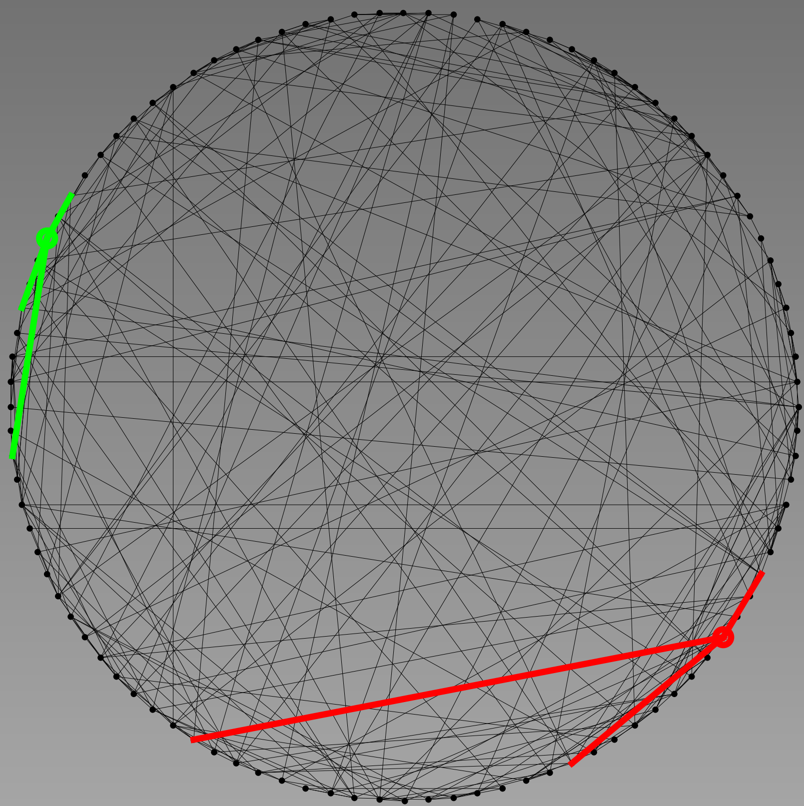
# The Method, cont.

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 $\ell_b$ as the product of all the lengths of their current connections. Then they calculate $\ell_a$ 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 $\ell_b$ as the product of all the lengths of their current connections. Then they calculate $\ell_a$ 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 $\ell_b/\ell_a$.

# 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:

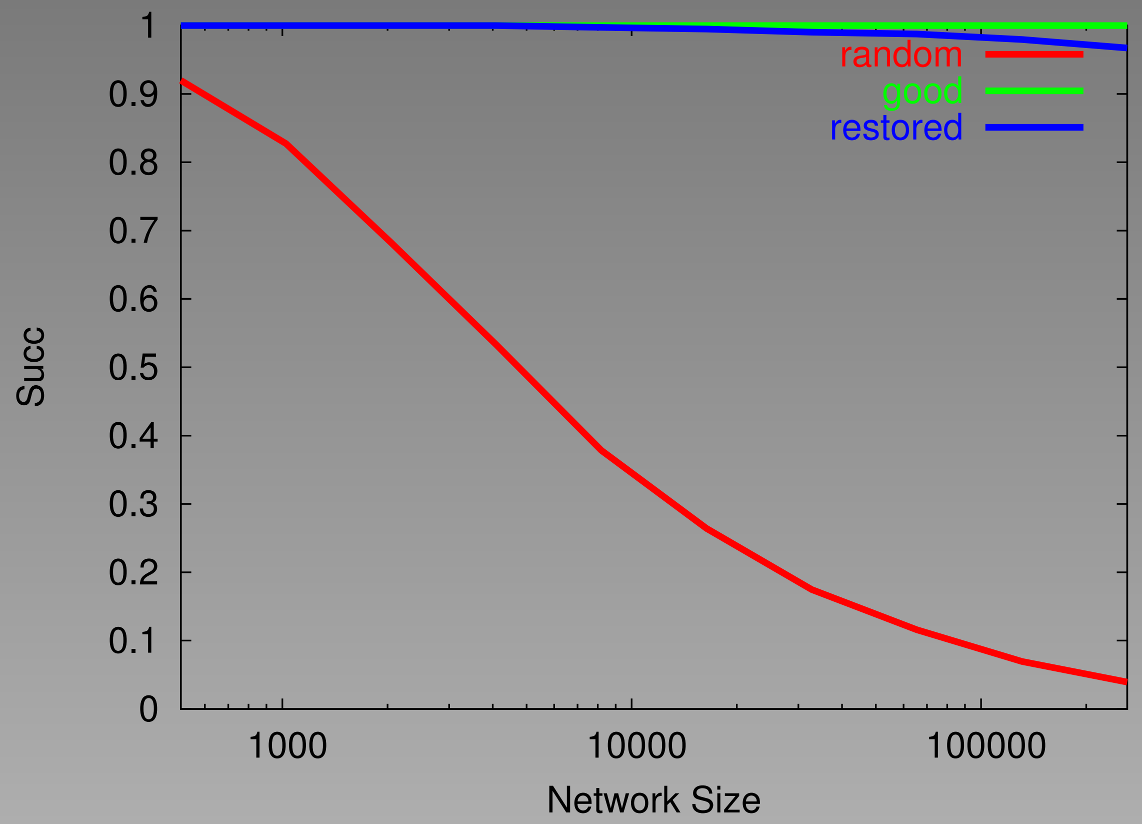
- Random walk search: “random”.
- Greedy routing in Kleinberg’s model with identities as when it was constructed: “good”.
- Greedy routing in Kleinberg’s model with identities assigned according to our algorithm (2000 iterations per node): “restored”.

# 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:

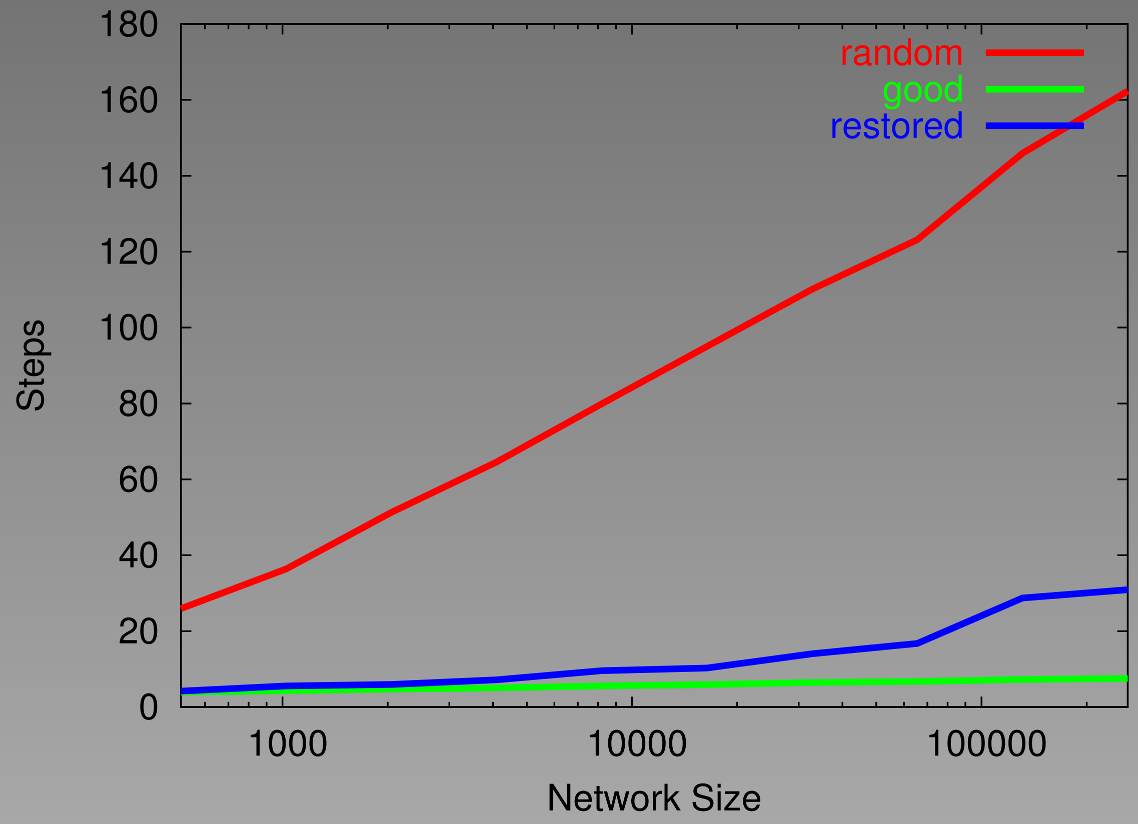


# Simulations, cont.

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.

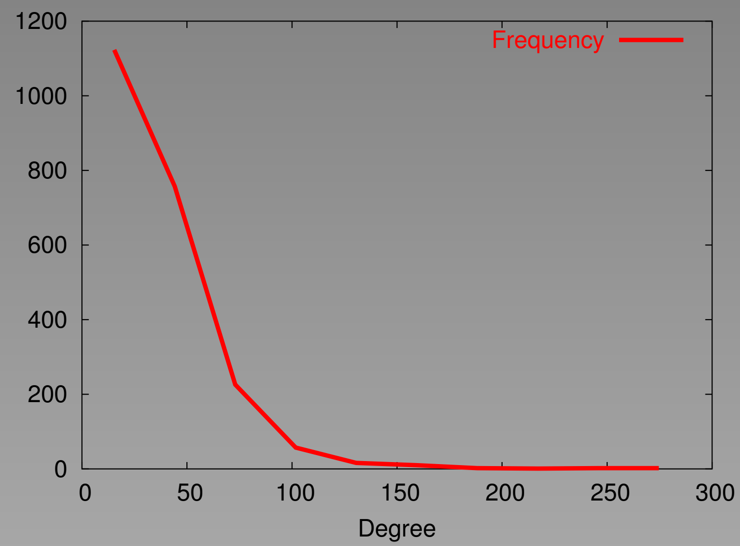
- The set was spidered so as to be comparatively dense (average 36.7 connections per person).

# 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. (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. (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.

# Further Research

- Will fitting the data against other models work better?

# Further Research

- Will fitting the data against other models work better?
- Can we find better selection functions for whom to switch with? (Gibbs type samplers.)

# Further Research

- Will fitting the data against other models work better?
- Can we find better selection functions for whom to switch with? (Gibbs type samplers.)
- What can we say about the convergence rate?

# Further Research

- Will fitting the data against other models work better?
- Can we find better selection functions for whom to switch with? (Gibbs type samplers.)
- What can we say about the convergence rate?
- It needs to be tested on more data.