Provable Polylog Routing for Darknets

Stefanie Roos Thorsten Strufe
TU Darmstadt, Germany
<lastname>@cs.tu-darmstadt.de

***Abstract*—Darknets, anonymous and membership-concealing P2P networks, aim at providing censorship-resistance without relying on a central authority. An efficient routing algorithm is needed to create Darknets that offer an acceptable performance to a large number of users. Designing such an algorithm is hard due to the restricted topology of Darknets, which has not been modelled adequately up to now. We present such a model of Darknets by modifying Kleinberg's small-world model [1] and a new algorithm, NextBestOnce. It is shown analytically that NextBestOnce takes $\mathcal{O}(\log^2 n)$ steps on our model, simulations show that it performs better than existing Darknet routing algorithms such as the one used in the dark Freenet [2], especially with regard to the maximal path length which is bounded by $\mathcal{O}(\log^2 n)$ for NextBestOnce, but scales linearly in case of Freenet.**

***Index Terms*—Darknets, P2P, Formal Models, Privacy**

I. Introduction

Fully privacy preserving and censorship resistant social networking, publication and communication, can be achieved in Darknets: P2P overlays in which only devices of individuals with a mutual trust relationship in the real world establish connections. Adversaries with the ability to resolve network addresses to the name and address of the individual using it are capable of persecuting participation in conventional overlays, which don't conceal the participation of members. Anonymization on lower layers, like for instance with onion routing, is detected comparably simply, and connections to known Tor nodes can be blocked easily. In Darknets, with lack of the freedom to establish connections between arbitrary nodes, routing topologies like in conventional P2P overlays can not be created, and efficient routing hence can not simply be achieved.

Several prior approaches have been proposed to implement Darknets or similar anonymous communication systems. Turtle [3] floods messages, MCON [4] reconstructs a routing topology by establishing tunnels between neighboring nodes in the identifier space, yet requires a globally trusted authority for joining, the existence of which may not be a realistic assumption. X-Vine [5] is a similar system, not relying on a trusted global authority, but as in case of MCON tunnels through trusted links, need to be rebuild in case of churn. The overhead of this in a real-world scenario is unclear, but is likely to be high and significantly reduce the performance. Various routing schemes have been deployed in Ad-hoc networks, e.g. Face routing [6], but an analytical analysis is missing. OneSwarm [7], finally, combines flooding with probabilistic routing to enhance anonymity and relaxes the requirement of full membership concealment by relying on links between devices of users without mutual trust. The amount of messages increases at least linearly with the system size for all those approaches. The only prior work attempting to fully meet Darknet requirements and achieve efficient routing is the dark Freenet [2]. While the actual number of Freenet users is unknown due to its membership-concealing nature, a report [8] has rated it the most popular and secure anti-censorship system in China. In particular it seems to be used more frequently than Tor.

Freenet attempts to recreate a ring as a routing topology by applying a Metropolis-Hastings algorithm to adapt node identifiers. The goal is to approximate a network topology that resembles Kleinberg's small world model, on which greedy routing is shown to converge within $\mathcal{O}(log^2 n)$ steps. However, the model assumes that every node is connected to its predecessor and successor on the ring, which cannot be achieved under Darknet constraints. Greedy routing hence is exchanged for a *distance-directed depth first search*, which, in this context, is not formally analyzed.

In this paper we generalize Kleinberg's model by relaxing the connections with direct neighbors in the identifier space to connections to nodes within a given distance C in the neighborhood, making the model more realistic for modeling Darknets. We then propose a class of routing algorithms, NextBestK, parametrized by k, which restrict the number of times the algorithm is allowed to increase the distance to the destination. The case $k = 1$, called NextBestOnce, is analyzed on the generalized model. We show analytically that the number of routing steps is in $\mathcal{O}(\log^2 n)$. Simulations confirm this result and show that NextBestOnce performs better than Freenet routing with regard to the average and maximal number of steps.

Section II provides some background on Kleinberg's model and search algorithms. Section III introduces NextBestK, followed by a theoretical analysis of NextBestOnce in Section IV. Simulations, which compare various routing algorithms on the generalized Kleinberg model, are presented in Section V, before concluding in Section VI.

II. Background

In this Section a formal definition of a routing problem is given. Furthermore Kleinberg's small world model [1] and its application to Darknet routing is discussed.

Routing Problem Definition

Routing on a graph $G = (V, E)$ is defined as the problem of finding a graph traversal from some node $s \in V$ to another node $t \in V$, or simply to determine a route from s to t. On

arbitrary graphs any routing algorithm has complexity $\mathcal{O}(|V|+|E|)$. Identifier assignment can be leveraged to design specific classes of graphs, e.g. small-world graphs with a low diameter, for which efficient routing algorithms can be provided. Such algorithms need to locally and recursively choose the next hop for the routes, to achieve membership-concealment and anonymity as required in Darknets.

Given a class of graphs $K(V)$ with node set V and edges chosen according to a random variable X, equipped with a distance function $dist : V \times V \to [0, \infty)$, we want to find an algorithm A with minimal expected cost over all instances in $K(V)$, i.e., find A, so that

$$\max_{s,t \in V} \mathbb{E}(R_A(s,t)) \tag{1}$$

is minimized, where $R_A(s,t)$ is the number of steps needed to find t from s. The quantity in Equation 1 is called the *greedy diameter* of A with respect to $K(V)$.

Routing in Small Worlds

The dark Freenet [2] is based on Kleinberg's small world model [1], in which $K(n,d)$ nodes are arranged in a d-dimensional lattice of side length n. Nodes are identified by an integer vector $v \in \mathbb{Z}_n^d$ representing their position on the lattice. The distance between two nodes v and w is the *Manhattan distance with wrap-around*:

$$dist(v,w) = \sum_{i=1}^{d} \min\{|v_i - w_i|, n + v_i - w_i, n + w_i - v_i\} \tag{2}$$

Each node v is given a long-range neighbor lr^v with

$$P(lr^v = w) = \frac{1}{K} \frac{1}{dist(w,v)^d} \tag{3}$$

where

$$K = \sum_{u \neq v} \frac{1}{dist(u,v)^d} = \Theta(\log n) \tag{4}$$

is a normalization constant.

Besides these long-range links each node is connected to all nodes within distance 1. This condition is essential for Kleinberg's proof that a standard greedy algorithm needs $\mathcal{O}(\log^2 n)$ steps. In the following, standard greedy refers to an algorithm that always chooses the neighbor closest to the destination given that is closer than the current node. Otherwise the algorithm terminates.

Kleinberg's model is indeed a small-world model, having a logarithmic diameter[9]. While the original model uses maximally three directed links, it can extended to power-law degree distributions and undirected long-rang links [**?**], [10]. Hence three characteristics of social networks- small-world, power-law degree distribution and clustering achieved by the distance distribution in the long-range link selection - can be incorporated into Kleinberg's model. The only problem with regard to Darknets is the presence of links between nodes that are closest in the ID space, which can in general not be guaranteed by an embedding into an euclidean space of bounded dimension. Both Sandberg [11] and Vasserman et al. [4] used graphs generated as in Kleinberg's model for their evaluation.

Sandberg [11] designed a Markov Chain algorithm for assigning IDs uniformly distributed between 0 and 1 to nodes such that the distribution of the distances between neighbors converges towards Equation 3. Standard greedy requires finding a hamiltonian cycle to work in a one-dimensional space, so it does not work for general graphs when choosing one-dimensional identifiers.

Sandberg considered two alternatives: *Backtracking* forwards the message to the predecessor of the current node, if no unvisited neighbor closer to the destination can be found. The algorithm terminates if either the destination is reached or the source has contacted all neighbors closer to the destination. The algorithm is more likely to succeed for an arbitrary pair of nodes, but it still needs a hamiltonian cycle (in case of a one-dimensional ID space) to succeed in case of nodes that are neighbors or close in distance. The second possibility is a *distance-directed depth first search*, where the neighbors are contacted in the inverse order of their distance to the destination. This is guaranteed to work on all connected graphs and is actually implemented in the dark Freenet [2], so we will simply refer to it as Freenet routing. Note, that neighbors far from the destination might be contacted. This makes it hard to analyze Freenet routing since the process made up to that point is annihilated.

III. NextBestK

We propose a class of algorithms, parametrized by $k \in \mathbb{N}$, which are a compromise between backtracking and Freenet. A node applying NextBest$K$ choses up to k neighbors in higher distance to the target than itself as next hops. A node v keeps track of the neighbors W_v it has already contacted and the predeccessor p_v. v is 'marked' in case it has contacted k nodes at a greater distance, meaning it should not be contacted anymore. Each node can determine if its neighbors are already marked. This can be realized by including a Bloom filter in the message, containing the identifiers of all 'marked' nodes. Note, that this is in agreement with Darknet constraints, since Bloom filter allow to check if a known identifier is contained in the Bloom filter, but do not allow to retrieve unknown identifiers.

Algorithm 1 describes one recursive step of the algorithm. The algorithm takes as input the predecessor of the current node, the target node, and the current node. In each non-terminal step of the algorithm, there are basically two possibilities: The node always contacts the neighbor closest to the destination, but if that neighbor is not closer than itself, the node is marked and will only be contacted by nodes to whom it forwarded the message. Besides the fact that a maximum of k neighbors in higher distance to the target can be chosen for each node, the algorithm has one fundamental difference to Freenet routing: nodes are not marked the first time they are contacted, and thus might have more than one predecessor (termination is still guaranteed, since in each step at least one edge is added to some set W_v and infinite loops are hence

avoided). The possibility to choose edges multiple times is essential for the complexity analysis, since it guarantees that in each step a maximal regression is not exceeded. Note, that in case $k = 1$, v does not need to keep track of W_v, since it is not contacted anymore afterwards. In this case, the overhead is identical to that of Freenet routing: one has to keep state only of the 'marked' nodes. In each step all neighbors are considered in both algorithm, NextBestOnce only needs an additionally comparison of the own distance to the destination to that of the closest unmarked neighbor. So the overhead of a step in NextBestOnce is barely higher than in Freenet routing. Backtracking in fact is the special case of NextBest0.

Algorithm 1 NextBestK(Node p, Node t, Node v)

```
    # INPUT: p predecessor, t target, v current node
    # N_v: neighbors of v
    # W_v: contacted neighbors u with dist(u,t) >= dist(v,t)
  if v == t then
    routing successful; terminate
  end if
  if v.predecessor == null then
    v.predecessor = p;
  end if
  S = {u in N_v :!u.marked AND u notin W_v}
  if S NOT EMPTY then
    nextNode = argmin_{u in S} dist(u,t)
    if dist(nextNode,t) >= dist(v,t) then
      W_v.add(nextNode)
      if |W_v| >= k then
        v.marked = true
      end if
    end if
  else
    v.marked = true
    nextNode = v.predecessor;
  end if
  if nextNode != null then
    NextBestK(v, t,nextNode)
  else
    routing failed; terminate
  end if
```

IV. Analysis of NextBestOnce

In the context of the generalized Kleinberg $\tilde{\mathcal{K}}(n, d, C)$, we analyze NextBestOnce, i.e., NextBestK for $k = 1$. $\tilde{\mathcal{K}}(n, d, C)$ relaxes the Kleinberg's requirement for *local edges*. While Kleinberg's proof requires that every node is connected to all nodes within distance 1, in $\tilde{\mathcal{K}}(n, d, C)$ nodes have neighbors within distance $C \geq 1$ in all directions. Due to space constraints and for clarity, we decided to use the simpler case of a directed graph with one outgoing long-range link per node. All the proofs presented in this Section work analogously for undirected graphs with a scale-free degree sequence, as is shown in [10]. This Section first introduces our Darknet model, then states our main result about NextBestOnce. In the context of that model routing with NextBestOnce from an arbitrary source node s to any destination node t requires $\mathcal{O}(\log^2 n)$ steps in expectation, the expectation is taken over all connected $g \in \tilde{\mathcal{K}}(n, d, C)$. The second part of this Section is dedicated to the proof of this result, which is split in two phases. The first phase treats the case when two nodes are at a distance that exceeds $K * \log^2 n$ for some constant $K > 0$. It is very similar to Kleinberg's original proof, with the additional difficulty that an improvement towards the destination is not guaranteed in every step. The idea is this step is to show that with a high probability the algorithm does not reach a vertex at twice the distance of the vertex closest to the destination. Hence the current distance does maximally have to be quartered to reach a node at half the distance of the closest node to the destination found up to that point. The second phase consists of covering the remaining distance in $\prime(\log^{\in} \setminus)$ steps. This is done by first showing that between any two nodes within distance C path of maximally length $\prime(\log^{\in} \setminus)$ exist with high probability, and then showing that NextBestOnce actually finds those paths.

Modelling Darknets

A graph $G \in \tilde{\mathcal{K}}(n, d, C)$ is created as follows:

We chose the node set V like Kleinberg with a distance function $dist$ as in Equation 2 and one long-range link per node randomly chosen as in Equation 3.

Each node $v = (v_1, ..., v_d)$ is given local links to neighbors above and below in terms of the identifier space: $a_1^v, ..., a_d^v, b_1^v, ..., b_d^v$,. Here a_j^v is chosen from the set

$$A_j^v = \{u = (u_1, ..., u_d) \in V : u_i = v_i \text{ for } i \neq j, \\ 1 \leq \min\{u_j - v_j, n + u_j - v_j\} \leq C\}$$

where $C \geq 1$ is a constant independent of n. Analogously b_j^v is chosen from

$$B_j^v = \{u = (u_1, ..., u_d) \in V : u_i = v_i \text{ for } i \neq j, \\ 1 \leq \min\{v_j - u_j, n + v_j - u_j\} \leq C\}.$$

For the proof we assume that the neighbors are chosen uniformly at random from A_j^v, respectively B_j^v, but this is only to minimize the number of variables for a clearer presentation. It works analogously if each element in the set is chosen with an arbitrary non-zero probability.

Local edges are considered undirected, while long-range links are directed as in the original model.

Main result

In the subsequent paragraphs we prove the following theorem using the above model for $d = 1$, which can easily be extended to any dimension d.

Theorem IV.1. *Let $R(s, t)$ be the number of steps that the algorithm NextBestOnce needs to find a path from s to t. For a connected graph $G = (V, E) \in \tilde{\mathcal{K}}(n, 1, C)$, $C \geq 1$ and arbitrary $s, t \in V$:*

$$\max_{s,t \in V} \mathbb{E}(R(s, t)) = \mathcal{O}(\log^2 n) \tag{5}$$

We begin by showing two basic facts about NextBestOnce. The proof of the routing performance is then divided in 2 phases. In the first phase one considers the case when the distance between the source s and the destination t exceeds $\mathcal{O}(\log^2 n)$, the second phase deals with the remaining distance of $\mathcal{O}(\log^2 n)$. The number of steps needed to complete the first phase is denoted by $R_1(s,t)$, the number of steps of the second phase $R_2(s,t)$.

Properties of NextBestOnce

The following observation is an essential property of NextBestOnce, that distinguishes it from Freenet routing and is the key to the theoretical analysis.

Observation IV.2. *For the algorithm NextBestOnce, the maximal increase of the distance to the destination is C for any graph $G \in \tilde{\mathcal{K}}(n,1,C)$.*

Obviously each node u has a local link to a node v, so that the $dist(u,t) < dist(v,t) \leq dist(u,t) + C$. Since the local links are considered undirected, v is not yet marked, because a node is only marked after all neighbors that are closer to the destination, which includes u, have been marked. Hence NextBestOnce can always choose a successor within distance C.

Lemma IV.3. *NextBestOnce terminates in $\mathcal{O}(Cn)$ steps for an arbitrary graph assuming that the distance to the destination increases by maximally C in each step.*

Proof: Assume the algorithm produces a circle of length l. Then at least $1/C * l$ nodes on the circle are declared marked. To see this, consider that the distance can only increase if a node is declared marked. If the sum of the total increase of a path equals the total decrease, one needs to increase the distance in at least $1/C$ of all steps, since the minimal decrease is 1 and the maximal increase is C. The algorithm ends after every node has been declared marked. The maximal number of steps decreasing the distance is less or equal to C times the number of increasing steps additional to the maximal distance of two nodes. So the total number of steps is bounded by $2n + Cn = \mathcal{O}(n)$ steps. ∎

In case that the maximal increase to the destination is restricted by a parameter C, independent of n, the maximal number of steps is linear in the network size. For arbitrary graphs, the algorithm terminates after $\mathcal{O}(n^2)$ steps by the above Lemma.

Phase 1

The following Lemma is essential for showing that the message gets within distance $\mathcal{O}(\log^2 n)$ of t in $\mathcal{O}(\log^2 n)$ steps. It is used to determine the expected improvement in each step.

Lemma IV.4. *Let $u, v \in V$ be arbitrary with $dist(u,v) = \Omega(\log^2 n)$. Then:*

$$P\left(dist(lr^u, v) \leq \frac{dist(u,v)}{4}\right) = \Omega\left(\frac{1}{\log n}\right).$$

Proof: For the first part, recall that there are $dist(u,v)/2$ nodes within distance $dist(u,v)/4$. The probability to have a link to one of them is at least $\frac{1}{K}\frac{1}{dist(u,v)*(1+1/4)} = \Theta(\frac{1}{dist(u,v)\log n})$ since $K = \Theta(\log n)$ by Equation 4. Multiplying with the number of nodes shows the claim. ∎

In the following consider two random processes $X_1, X_2, ...$, the distance to the destination of the i-th node on the route, and $Y_1, Y_2, ...$ with $Y_i = \min\{X_1, ..., X_i\}$.

Lemma IV.5. *Let $s, t \in V$. Then the expected number of steps NextBestOnce needs to reach a node within distance $\mathcal{O}(\log^2 n)$ of t is $\mathbb{E}(R_1(s,t)) = \mathcal{O}(\log^2 n)$.*

Proof: The idea of the proof is to show that in expectation it takes $\mathcal{O}(\log n)$ steps to half the distance to the destination. Then, to get within distance $\mathcal{O}(\log^2 n)$ the distance needs to be halved $\mathcal{O}(\log n)$ times. Combining the two gives the bound $\mathcal{O}(\log^2 n)$.

Formally, let $S(Y)$ be the number of steps needed to get within distance $Y_i/2$ of t when at distance Y. We want to show that $\mathbb{E}(S(Y_i)) = \mathcal{O}(\log n)$ for $Y_i > C\log^2 n$. This is done by considering two cases, $X_i \leq 2 * Y_i$ and $X_i > 2 * Y_i$, i.e.,

$$\begin{aligned}\mathbb{E}(S(Y_i)) = &\mathbb{E}(S(Y_i)|X_i \leq 2 * Y_i)P(X_i \leq 2 * Y_i) \\ &+ \mathbb{E}(S(Y_i)|X_i > 2 * Y_i)P(X_i > 2 * Y_i)\end{aligned}$$

By Lemma IV.4 $\mathbb{E}(S(Y_i)|X_i \leq 2 * Y_i) = \mathcal{O}(\log n)$ and by Lemma IV.3 $\mathbb{E}(S(Y_i)|X_i > 2 * Y_i) = \mathcal{O}(n)$. It remains to show $P(X_i > 2 * Y_i) = \mathcal{O}(1/n)$ to get:

$$\mathbb{E}(S(Y_i)) = \mathcal{O}((1 - 1/n)\log n) + \mathcal{O}(1/n * n) = \mathcal{O}(\log n)$$

Now if $X_i > 2 * Y_i$, the distance has to increase at least Y_i/C times, since the maximal increase is C. By Lemma IV.4 the probability that the distance is reduced to a fourth is at least $\Omega(\frac{1}{\log n})$ in each step. The probability that this does not happen in Y_i/C steps is $\mathcal{O}\left((1 - \frac{1}{\log n})^{Y_i/C}\right)$. Recall that $Y_i \geq C * \log^2 n$, and for $0 \leq p \leq 1$, $p^x \leq p^y$ if $x \geq y$. Then the probability is bounded from above by:

$$\mathcal{O}\left((1 - \frac{1}{\log n})^{C\log^2 n/C}\right) = \mathcal{O}\left(e^{-\log n}\right) = \mathcal{O}\left(\frac{1}{n}\right)$$

This completes the proof. ∎

Phase 2

For the second phase of the routing some notation is needed. Let $u_0 \leftrightarrow u_{m+1}$ denote the fact that nodes u_0 and u_{m+1} are connected by a path $u_0, ..., u_{m+1}$ and $dist(u_i, v)$ is monotone decreasing. We first determine the probability that two randomly chosen nodes are connected in such a way by a path of local links.

Lemma IV.6. *It is $u \leftrightarrow v$ with probability of at least $1/C$ for arbitrary $u, v \in V$.*

Proof: The Lemma is shown by induction on the distance $dist(u,v)$. If $dist(u,v) \leq C$, $u \leftrightarrow v$ holds with probability at least $1/C$. Otherwise assume the claim holds for all pairs u, v

with $dist(u,v) \leq K$. Let u and u' have distance $K+1$, where $K+1$ is less than $n/2$. Then u' is connected by a local link to at least one node v with $K+1-C \leq dist(u,v) \leq K$. The probability that $u \leftrightarrow v$ is $1/C$. Hence the probability $u' \leftrightarrow u$ is at least $1/C$ as well. The distance property is clear from the construction. ∎

For the second phase of the routing to be successful in a polylogarithmic number of steps, we need to show that there are short paths between nodes within a small distance.

Lemma IV.7. *For one pair of nodes u, v with $dist(u,v) \leq C$ there is a node w,$u \leftrightarrow w$ and $w \leftrightarrow v$, so that $dist(u,w) = \mathcal{O}(\log^r n)$ for any $r > 1$ with probability approximately $1 - 1/n^{\log^{r-1} n/C^2}$.*

Proof: For any node w the probability that $u \leftrightarrow w$ and $w \leftrightarrow v$ is at least $1/C^2$ by Lemma IV.6. Hence the probability of the complement of u and v being connected to one node within distance $\log^r n$ is at most $\left(1-1/C^2\right)^{\log^r n} \approx e^{-\log^r n/C^2} = n^{\log^{r-1} n/C^2}$. ∎

This is the main component for showing that any nodes within distance $\mathcal{O}(\log^2 n)$ steps can be found in $\mathcal{O}(\log^2 n)$ steps.

Lemma IV.8. *For arbitrary $G = (V,E) \in \tilde{\mathcal{K}}(n,1,C)$, let $s,t \in V$ be within distance $\mathcal{O}(\log^2 n)$, then $\mathbb{E}(R_2(s,t)) = \mathcal{O}(\log^2 n)$.*

Proof: By following a path of length at most $\mathcal{O}(\log^2 n)$ one ends at some u with $dist(u,t) \leq C$. Applying Lemma IV.7 with $r = 2$, w.h.p. u and t are connected via a path of length $\mathcal{O}(\log^2 n)$. So t can be found by only routing on a subgraph of size $\mathcal{O}(\log^2 n)$. Note, that the maximal increase to the destination is C. Hence, by Lemma IV.3, the worst-case performance of NextBestOnce for the second phase is $\mathcal{O}(C \log^2 n) = \mathcal{O}(\log^2 n)$ given that s and t are connected by a path of of length $\mathcal{O}(\log^2 n)$. If that is not the case the worst-case performance is $\mathcal{O}(n)$, by Lemma IV.3. Hence:

$$\mathbb{E}(R_2(s,t)) = \mathcal{O}(\log^2 n) + 1/n^{\frac{\log n}{C^2}}\mathcal{O}(n) = \mathcal{O}(\log^2 n)$$

∎

The proof of Theorem IV.1 follows from combining the two phases:

Proof: Let $s,t \in V$ be arbitrary. Let $R_1(s,t)$ denote the number of steps needed to get within distance $\mathcal{O}(\log^2 n)$ of t, and $R_2(s,t)$ be the number of steps needed to overcome the remaining distance. By Lemma IV.5 and Lemma IV.8:

$$\begin{aligned}&\mathbb{E}(R(s,t)) = \mathbb{E}(R_1(s,t)) + \mathbb{E}(R_2(s,t))\\ &= \mathcal{O}(\log^2 n) + \mathcal{O}(\log^2 n) = \mathcal{O}(\log^2 n)\end{aligned}$$

Since s,t are chosen arbitrarily, we also have

$$\max_{s,t \in V} \mathbb{E}(R(s,t)) = \mathcal{O}(\log^2 n).$$

∎

Note, that NextBestK would need k local links in each direction, so that a minimal increase of C in the distance to the destination can guaranteed. The proof then works analogously.

V. Simulations

Several routing algorithms are simulated on synthetic friendship graphs generated according to the model from Section IV for various graph sizes and values for C. Success rate, average, and maximum number of routing steps are measured. A high success rate guarantees the user that his request are answered. The average and maximal path length are important, since in most scenarios, the time to wait for an answer is of relevance. The average path length is more important in most cases, however having a bound on the maximal path length gives a guarantee on how long a user has to wait in the worst-case.

We concentrated our evaluation on synthetic graphs, since any real world graphs require the use of an embedding, and the routing would be depend on the properties of this embedding.

Setup

The model is implemented in GTNA [12]. First, a graph $G \in \tilde{\mathcal{K}}(n,1,C)$ for the desired $n, C \in \mathbb{N}$ is constructed as described in Section IV. Then the performance of a routing algorithm A is tested as follows: Each node s is chosen as the source for five distinct queries, i.e., the routing algorithm A is applied to find a path from s to five randomly chosen targets $t \in V \setminus \{s\}$. The success rate, average, and maximum number of hops then are measured based on these $T = 5 * n$ queries. The metrics are defined as follows: Let Q_A be the set of successful queries for routing algorithm A and let $L_A(q)$ denote the number of steps needed for $q \in Q_A$. Then the *success rate (recall)* of A is defined as $S(A) = \frac{|Q_A|}{T}$, the *mean number of steps* as $M(A) = \sum_{q \in Q_A} L_A(q)/|Q_A|$ and *maximal number of steps* as $Max(A) = \max_{q \in Q_A}\{L_A(q)\}$. Knowledge about the marked flags of nodes is assumed to be available to a node without additional messaging. Incorporating Bloom filters into each message that contain all nodes on the path is one possible solution that adheres to the Darknet restrictions and provides this information.

Eight routing algorithms are compared: Standard greedy, Backtracking, Freenet and NextBestK for $k = 1,2,3,10,\infty$. The parameter C is varied between 1,5 and 10. Graphs between 1000 and 100,000 nodes are analyzed, starting with 1000, 10,000 and then increasing by steps of 10,000. Due to space restrictions tables only show the values for 1000, 10,000 and 100,000. Averages are taken over 100 runs. Tables and Figures display means and standard deviation.

Success Rate (Recall)

The success rate is evaluated both without TTL and - in case of Freenet and NextBestOnce - for various TTLs in the order of $\log^2 n$. Without TTL, Freenet and NextBestK are expected to achieve a success rate of 100 % on our model. The success rate of greedy routing is 100 % for $C = 1$, but expected to fall with rising C. Backtracking should perform better, but fail for queries between nodes that are close in distance, since for those the probability that a path along which the distance decreases in every step is lower. The success rate of Backtracking is likely to be influenced by both C and n.

As for the TTL, by the analysis in Section IV, NextBestOnce should achieve a success rate close to 100 % for $c * \log^2 n$ for some constant $c > 0$, while Freenet routing is expected to have a lower success rate for such a TTL.

Our measurements show that the success rate of standard greedy is low, 41.5 to 43 % for $C = 10$ and 64 to 65 % for $C = 5$. The success rate of Backtracking is much higher, with values from 98% for a graph with 1000 nodes and $C = 10$ to 99.9% for a graph of $100,000$ nodes with $C = 5$.

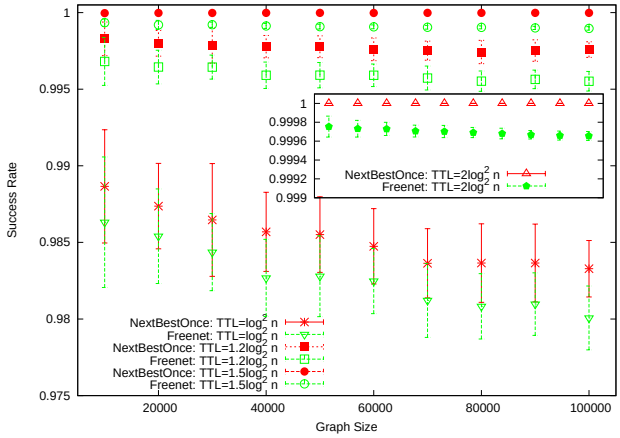

Fig. 1. Graph Size vs. Success Rate for various TTL

Freenet and NextBestK always terminate successfully when using our model. Experimenting with various TTLs shows that for a TTL of $\log^2 n$ over 98% of the queries are successful. Figure 1 shows the success rate for graphs of size 1,000 to 10,000 with $C = 5$ and TTL $\log^2 n$, $1.2\log^2 n$, $1.5\log^2 n$ (main plot) and $2\log^2 n$ (right). For a TTL of $1.5\log^2 n$, both algorithms achieve a success rate of nearly 100 %. Looking at $TTL = 2\log^2 n$ in the right of Figure 1 in more detail, one can distinguish that NextBestOnce indeed has a success rate of 100 %, while Freenet fails for some queries. The variance drops with the graph size, which can be explained by the fact that T, the number of executed queries, increases linearly with n. The difference in success between NextBestOnce and Freenet is small, however, NextBestOnce outperforms Freenet for any TTL in the order of $\log^2 n$ for all considered graphs.

Routing Length

Routing Length Average and maximum number of hops without TTL are evaluated for all simulated algorithms. Due to space restrictions, results for the average number of steps are presented for Freenet and NextBestK, the algorithms with a success rate of 100 %. The maximum number of steps is compared for Freenet and NextBestOnce. NextBestOnce is expected to have lower average and maximum numbers of steps, since per analysis above it is in $\mathcal{O}(\log^2 n)$. Freenet has not been formally analyzed, since the regression in the distance is unbounded, which complicates the analysis. NextBestK for $k > 1$ has the same drawback in the considered model, since only a single neighbor in higher distance to the target is guaranteed to be a local neighbor. NextBestK additionally allows using edges several times, in the worst case only marking one edge in a circle, so the worst-case performance is $\mathcal{O}(|E|n)$. For this reason NextBestK for $k > 1$ is expected to perform worse than Freenet.

n	Mean Steps	Max Steps
1000	23.34 ± 0.78	72.32 ± 6.99
10000	45.60 ± 0.67	153.44 ± 10.90
100000	76.33 ± 0.50	265.93 ± 13.05

TABLE I
STANDARD GREEDY ON KLEINBERG'S MODEL

n	C	Freenet	NextBestOnce	NextBest2
1000	5	17.93 ± 0.61	17.97 ± 0.62	18.79 ± 0.74
	10	16.84 ± 0.44	17.20 ± 0.45	19.04 ± 0.54
$10,000$	5	38.05 ± 0.72	37.77 ± 0.56	38.73 ± 0.7
	10	36.05 ± 0.72	35.98 ± 0.66	38.39 ± 0.68
$100,000$	5	67.87 ± 1.37	65.94 ± 0.51	67.2 ± 0.52
	10	65.56 ± 1.43	62.99 ± 0.44	66.35 ± 0.53
		NextBest3	NextBest10	NextBestAll
1000	5	18.94 ± 0.72	19.68 ± 0.78	19.73 ± 0.79
	10	19.99 ± 0.62	22.13 ± 0.67	22.28 ± 0.7
$10,000$	5	39.55 ± 0.71	40.64 ± 0.7	40.89 ± 0.78
	10	40.24 ± 0.69	44.71 ± 0.78	44.82 ± 0.77
$100,000$	5	68.28 ± 0.48	70.59 ± 0.57	70.57 ± 0.57
	10	69.43 ± 0.56	77.47 ± 0.50	77.56 ± 0.55

TABLE II
AVERAGE NUMBER OF STEPS

Note, that for $C = 1$ all the considered algorithms perform like greedy, because a node closer to the target is found in each step. Hence it is only necessary to compute the metrics for greedy. Table I displays the average and maximal number of step for graphs of 1000, 10,000 and 100,000 nodes. Both increase by a factor of less than 4 from 1000 to 10,000 nodes. The maximum is about three to four times the mean. The performance of Freenet and NextBestK with regard to the average number of steps complies with the expectations: NextBestK for $k > 1$ performs worse than Freenet, the average path length increasing with k. Table II shows that in case of 1000 nodes Freenet has an average path length of 16.84 for $C = 10$, while NextBestAll, i.e., the case $k = \infty$, needs 22.28 steps. For 100,000 nodes the respective values are 65.65 and 77.56. An interesting difference is that in case of Freenet, NextBestOnce, and to some extent NextBest2 the number of steps decreases with C, while they increase for the other algorithms. This can be explained since the decrease of the distance when using a local link increases with C. However, a higher value for C means that the last phase of the routing, finding the target using local links, lasts longer, because closeness is not a good indicator to find the correct route anymore. NextBestOnce and Freenet seem to be able to deal with this problem rather well, while the other do not. This is probably due to the fact that a lot of edges are used several times, i.e., the routing path contains a lot of circles.

Freenet and NextBestOnce perform very similar in graphs of size 1000 and 10,000 (differences vary between 0.1 and 0.7 steps), NextBestOnce is slightly but clearly better for 100,000 nodes, needing only 63 steps in comparison to 65.6 in for $C = 10$. This, too, is in line with our expectation. The comparison of the maximal number of steps in Table III reveals that there are indeed some queries for which the Freenet algorithm is unsuitable, needing as many steps as 1.5 times the graph size. In case of NextBestOnce the maximal path length is about three to four times the average number of steps, which is similar to the case of standard greedy with $C = 1$. In case of Freenet routing the maximal path length grows at least linearly with the network size.

n	C	Freenet	NextBestOnce
1000	5	104.86 ± 187.94	57.31 ± 5.27
	10	158.24 ± 299.78	62.71 ± 6.55
$10,000$	5	4771.72 ± 7327.84	127.02 ± 10.97
	10	5158.67 ± 7253.70	127.55 ± 9.79
$100,000$	5	158777.13 ± 65702.56	236.08 ± 14.28
	10	174578.83 ± 55740.99	225.66 ± 13.43

TABLE III
MAXIMUM NUMBER OF STEPS

Summary

The simulations give four important insights: First, the trivial result that standard greedy does only work in a relatively low percentage of cases. Secondly, NextBestK is not suitable for routing on graphs with less than k local links in each direction. Furthermore, the simulations of NextBestOnce agree closely with the theoretical result from Section IV indicating a polylogarithmic performance. Finally, with regard to the maximal number of routing steps NextBestOnce outperforms Freenet routing drastically.

VI. CONCLUSION

We have introduced NextBestK, a class of Darknet routing algorithms. Generalizing Kleinberg's model to relaxed neighborhood connectivity, which better represents graphs that can be approximated in Darknets, we analyzed their performance. NextBestOnce, an instance from this class, is shown to converge in $\mathcal{O}(log^2 n)$ steps. Our simulations, comparing NextBestK with the conventional Freenet routing, confirm that both the average as well as the maximal path length are polylogarithmical for NextBestOnce, while Freenet routing exhibits linear growth in the network size.

In prior work we analyzed the topology approximation of Freenet and have proposed a more resistant embedding [13]. Combining the two we are expecting to be able to provide a novel Darknet that is both efficient, and resistant towards adversarial behavior. We applied more realistic degree sequences to the extended model in [10] and we expect to be able to use it for further analysis of Darknets, or similar Adhoc net routing algorithms.

REFERENCES

[1] Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Symposium on Theory of Computing*, 2000.

[2] Ian Clarke et al. Private communication through a network of trusted connections: The dark freenet. http://freenetproject.org/papers.html,10-12-2010.

[3] Bogdan C. Popescu, Bruno Crispo, and Andrew S. Tanenbaum. Safe and private data sharing with turtle: Friends team-up and beat the system. In *Workshop on Security Protocols*, 2004.

[4] Eugene Vasserman et al. Membership-concealing overlay networks. In *CCS*, 2009.

[5] Prateek Mittal, Matthew Caesar, and Nikita Borisov. X-vine: Secure and pseudonymous routing using social networks. *CoRR*, 2011.

[6] Brad Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. MobiCom '00.

[7] Tomas Isdal et al. Privacy-preserving p2p data sharing with oneswarm. In *SIGCOMM*, 2010.

[8] Freedom House. Country report for end users in China. http://www.freedomhouse.org/sites/default/files/LOtF_China.pdf.

[9] Chip Martel. Analyzing Kleinberg's (and other) small-world models. In *Proceedings of ACM Symp. on Princ. of Dist. Comp. (PODC*, pages 179–188. ACM Press, 2004.

[10] Stefanie Roos. Analysis of routing on sparse small-world topologies. Master's thesis, Technische Universität Darmstadt, 2011.

[11] Oscar Sandberg. Distributed routing in small-world networks. *Algorithm Engineering and Experiments*, 2006.

[12] Benjamin Schiller, Dirk Bradler, Immanuel Schweizer, Max Mühlhäuser, and Thorsten Strufe. GTNA: A Framework for the Graph-theoretic Network Analysis. In *Springsim*, 2010.

[13] Benjamin Schiller, Stefanie Roos, Andreas Höfer, and Thorsten Strufe. Attack resistant network embeddings for darknets. In *IEEE SRDS/WNR*, 2011.