Optimizing the Placement of Implicit Proxies

Jacopo Cesareo
jcesareo@cs.princeton.edu

Josh Karlin
jkarlin@bbn.com

Michael Schapira
schapiram@huji.ac.il

Jennifer Rexford
jrex@cs.princeton.edu

ABSTRACT

Traffic filters block clients from communicating with certain Internet destinations. To prevent clients from evading the filtering policies, traffic filters may also block access to well-known anonymizing proxies. In response, researchers have designed more sophisticated solutions techniques that rely on *implicit* proxies lying along the path to unfiltered destinations. An implicit proxy transparently deflects traffic directed to an unfiltered destination toward the filtered destination. However, the effectiveness of implicit proxies highly depends on their presence in paths between clients and unfiltered destinations. In this paper we formulate and solve the problem of proxy placement, and evaluate our algorithms on snapshots of the Internet topology for a variety of client and destination sets. We also consider smart filtering techniques that select alternate routes to avoid implicit proxies, as well as the effects of asymmetric Internet routing. Our results show that a relatively small number of proxies can satisfy a large group of clients across a range of geographic locations.

1. INTRODUCTION

Network service providers increasingly block, filter, redirect, intercept, or even modify traffic between their users and popular or controversial websites or other Internet-based services [1, 2, 3]. Most techniques to bypass such filters [4, 5, 6] rely on *explicit proxies*, where the clients send their packets to a public VPN server or an anonymizing proxy like TOR (The Onion Router) [5], which in turn directs traffic to the filtered destination. However, service providers can block access to these proxies simply by adding them to the list of filtered IP addresses. This forces the proxy services to change IP addresses (or even IP prefixes) frequently, in an ongoing cat-and-mouse game with the traffic filters.

Recently, researchers have utilized *implicit* proxies to avoid these problems by placing proxies on the path from the clients to seemingly innocuous destinations. However, the success of these techniques depends on the placement of implicit proxies at strategic locations that lie on many paths between clients and unfiltered destinations.

1.1 Decoy Routing

Implicit proxies are an effective way to offer services to clients without explicit configuration. Historically, service providers deployed implicit Web proxies at client access points, to serve cached content without requiring users to configure their browsers to use a proxy. Using implicit proxies to avoid traffic filters raises additional challenges. First, the proxies must be placed outside of the region controlled by the traffic filter, making it harder to ensure that client traffic traverses the proxy. Second, clients must simultaneously obscure the IP addresses of intended destinations (to evade the traffic filters), and signal the real addresses to the implicit proxy (to ensure the traffic reaches the intended destination).

During the past few years, three works have proposed effective ways to use implicit proxies to bypass traffic filters: Cirripede [7], Decoy Routing [8], and Telex[9]. Even though there are subtle differences between them, the projects share a common use case. A client accesses Internet services through a traffic filter that blocks access to the address of the intended destination. If the client tries to connect to the destination explicitly, the connection is blocked by the filter (1). To get through the filter, the client initiates a connection to a non-filtered destination address (2). In reality, this connection camouflages a signal with the intent of the client to connect to the implicit proxy. A router on the path of the flow detects the signal and redirects the flow to the implicit proxy (perhaps running directly on the router), which in turn directs the traffic to the intended destination (3). Cirripede, Decoy Routing and Telex have given different names to the non-filtered destination as well as the router-proxy combination. Without loss of generality, we use the nomenclature used in Decoy Routing: we refer to *decoy destinations* for non-filtered destinations and *decoy routers* (DR) for the traffic-deflecting components. We refer to the overall scheme as *decoy routing*.

The success of decoy routing rests on two conditions: 1) the traffic filter cannot distinguish the signal from legitimate traffic and 2) the DR lies on the path between client and decoy destination. The former condition is

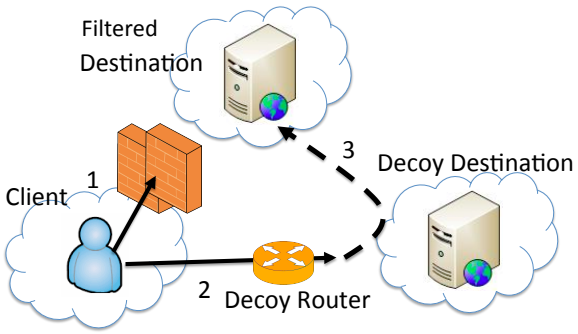


Figure 1: Decoy Routing Scheme

solved by injecting pseudo random values in traffic headers.[1] The latter generates the complex problem of *Decoy Router Placement*.

1.2 Optimizing Decoy Router Placement

Since clients do not send packets directly to the decoy router, traffic filters cannot easily block access. To block access, the service provider has to block all traffic that traverses the DR en route to decoy destinations. On the other hand, the DR must lie on the path between the client and the decoy destination. That is, effective DR placement is not just important for good performance—it is crucial for the solution to work in the first place. Placing DRs at many locations throughout the Internet, though, could be prohibitively expensive: the goal is to place them strategically to cover many filtered clients at minimal cost.

The decoy router placement (DRP) problem is the problem of placing decoy routers in the network to maximize the number of filtered clients that traverse DRs en route to decoy destinations. A 'naïve' solution would deploy a decoy router on each path between a client and a decoy destination. A more sophisticated solution would strategically place decoy routers at locations that appear on many paths, to maximize coverage with as few decoy routers as possible.

Previous research in the area has stopped short of exploring the optimal number and distribution of DRs. The work in Cirripede [7] touches on the subject suggesting that two Tier-1 Autonomous Systems (ASes) would need to be instrumented, but many questions remain about how to place decoy routers and how to handle smart traffic filters that attempt to avoid the decoy routers.

In this paper we make the following contributions:

- We formulate the DRP problem as a monitor placement problem and correlate the two problems by presenting effective heuristics to find efficient DRP and monitor placement solutions.
- We introduce a more challenging variant of the DRP problem, DRP against *smart* filtering (DRPSF), and show its complexity and inapproximability bounds.
- We evaluate the DRP and DRPSF problems on a wide range of real-world scenarios and show that efficient DR placement is indeed feasible no matter the clients, decoy destinations, filtering level or path properties.
- Lastly, we relate the results of the DRP problem to the structure of the network and conclude that the latter makes efficient placement possible.

[1]In Cirripede the signal is hidden within the initial sequence number of TCP SYN packets. In Decoy Routing and Telex the signal is the manipulation of the random nonce in the *Hello* packet in the TLS protocol. Specifically, in Decoy Routing the nonce is a shared secret between client and DR while in Telex it is the DR's public key.

2. DECOY ROUTER PLACEMENT

In this section we formulate the DRP problem (§2.1). Then, in § 2.2 and § 2.3, we propose two metrics to evaluate candidate solutions, and their respective algorithms for finding such solutions. We also report the theoretical results on the relationship between the algorithms' solutions and the optimal solution.

2.1 Decoy Router Placement Problem

We abstract the AS-level Internet as a graph $G = \{V, E\}$ of nodes V and edges E. Nodes $v \in V$ represent Autonomous Systems (ASes), and edges $e \in E$ represent logical connections between them. An AS, or domain, is a collection of devices under the control of a single entity. We model the components of the decoy routing scheme with respect to the AS-level graph G.

Clients and Decoy Destinations: Clients and Decoy Destinations are *nodes* in the graph. In particular, we define a set $C \subset V$ of clients and a set $D \subset V$ of decoy destinations. It is worth noting that in our model no node can act as both a client and decoy destination.[2]

Paths: Traffic flows from a node i towards a node j on *path* p_{ij}. Since traffic at the AS level is often asymmetric, path p_{ji} might not be the same as path p_{ij}. Initially, we make the assumption that there is a *single* path from client i to decoy destination j. We later consider the case where a client can choose to reach the decoy destination through a set of routes (thus enabling a smart filter to avoid the route with a DR). We define P as the set of all paths between clients in C and decoy destinations in D.

The Cirripede, Decoy Routing and Telex schemes differ in whether or not their decoy routers need to observe return traffic. If a decoy router is on path p_{ij} (client i to destination j) and p_{ji} then the router has more information and control over the flow. The decoy routers in the Telex [9] architecture must observe traffic

[2]$C \cap D = \emptyset$

Notation	Definition
C	set of Clients
D	set of Decoy Destinations
p_{ij}	Path from $c_i \in C$ to $d_j \in D$
P^i	set of Paths between $c_i \in C$ and D
P	set of Paths between C and D
R	DR Candidate Solution set
N_x	AS neighbor set of AS X
P^i_j	set of all valid paths between c_i and d_j

Table 1: Decoy Router Placement Notation

in both directions. We call this the *bidirectional* requirement. Cirripede and Decoy Routing [7, 8] only need to observe traffic on path p_{ij} (termed *unidirectional*). Clearly, the bidirectional requirement will reduce the number of available decoy destinations provided by each DR, as discussed in § 5.4.

Decoy Routers : We associate a DR with either a node or edge in the graph. Thus, our analysis will identify either entire ASes or individual inter-AS links to instrument with DRs. Identifying ASes to cover gives a bound on the number of individual organizations that would be needed to deploy decoy routers. Identifying inter-AS links to instrument is more fine grained and helpful in the event that the decoy router is simply a bump-in-the-wire device instead of an actual router. We define R as the set of candidate DR locations. We assume that a client node will not host DRs.[3]

The DRP problem is therefore equivalent to finding a set of nodes or edges R to **cover** paths in P. The goal is to find a solution R which covers P efficiently.

We give a specific formulation of the problem: given a fixed number of DRs k, what is the best placement solution R for these DRs? To evaluate a solution we propose two metrics: **fraction of pairs covered** and **fraction of α-covered clients**. We discuss these metrics in the following sections.

For clarity, the notation introduced so far is summarized in Table 1.

2.2 Fraction of Pairs Covered

The first metric we propose, fraction of pairs covered, evaluates the goodness of a solution R of size k by the fraction of client-decoy destination pairs covered in P. The fraction of pairs covered is equivalent to the average success rate of any client c to leverage the decoy routing scheme when picking a decoy destination d at random from set D.

Given this metric, the goal of the DRP problem is to maximize the fraction of pairs covered in P with a fixed number k of routers R:

$$max_{R:|R|=k}|\{p \in P|\exists r \in R \text{ s.t. } r \in p\}|$$

We refer to this problem as *FPC-k*.

[3] $R \cap C = \emptyset$

We point out that *FPC-k* can be formulated as the BCMCP (Budget Constrained Maximum coverage problem without sampling) problem proposed by Suh et al. in [10], where the goal is to maximize the number of traffic flows sampled with a fixed number of monitors (if we identify DRs with active monitors capable of sampling and redirecting traffic and we focus on flows between clients and decoy destinations to sample).

As with BCMCP, FPC-k is a NP-hard to approximate within a ratio better than $1 - \frac{1}{e}$, as it can be easily reduced to the MAX-k-COVER [11] problem. A matching $1 - \frac{1}{e}$ approximation guarantee is achieved by a simple greedy heuristic which, at each step, maximizes the utility value of the chosen element (monitor/DR). In the next section we define this greedy algorithm for the DRP problem.

We present a greedy algorithm with a (tight) $1 - \frac{1}{e}$ approximation ratio, which we will refer to as *GreedyPairs*.

2.2.1 GreedyPairs Algorithm

A greedy algorithm's trademark is to make the locally optimal choice in each of its iterations. For *FPC-k*, the local optimal choice is to pick the location that covers the largest number of (previously uncovered) $< c, d >$ pairs.

GreedyPairs starts by considering P as the set of *outstanding* paths OP, $OP = P$. Then, it iteratively picks the most popular location, updating OP accordingly. We summarize *GreedyPairs*'s steps in Alg. 1.

Algorithm 1 GreedyPairs

- Ranking: rank each location based on the number of paths in OP traversing it
- Greedy Choice: pick element x with highest rank (breaking ties arbitrarily). Add x to R
- Input Update : update outstanding paths by removing from OP all paths P_x containing element x.
- Termination: if $|R| = k$, stop. Otherwise, repeat from Step 1

See [12] for a proof that this greedy algorithm provides a $(1 - \frac{1}{e})$ approximation to the optimal solution of FPC-k.

2.3 Fraction of α-covered clients

The second metric we propose focuses on *client coverage*. Each client c_i connects to several, if not all, decoy destinations $d \in D$ through a set of paths P^i. $P^i \subset P$ is the set of all paths with source c_i. The key observation is that P^i does not need to be fully covered for a client to leverage decoy routing. Rather, an acceptable frac-

tion α of P^i covered guarantees the client a good chance of leveraging decoy routing. For example, for $\alpha = 0.5$, the client has a fifty percent chance of successfully leveraging the decoy routing scheme when randomly picking a decoy destination. Thus, we consider a client to be covered if fraction α of its pairs are covered.

Through the notion of α-covered clients, we evaluate the goodness of a solution R as the **fraction of α-covered clients** (FαCC). Similarly to §2.2, we can formulate the DRP problem as the problem to maximize the number of α-covered clients given a fixed solution size k:

$$max_{R:|R|=k}|\{c \in C|\text{C is } \alpha - \text{covered by R}\}|$$

We refer to this problem as *FαCC-k*. Unlike *FPC-k*, we note that this variant formulation does not have a monitor placement equivalent. The unique characteristics of the DR scheme, the need to provide a client a reasonable probability of success, are different from the needs of classic monitor placement problems.

We relate the complexity of this formulation to the DENSEST-SUBGRAPH problem [13, 14]. An approximation preserving reduction from DENSEST-SUBGRAPH to *FαCC-k* (when $\alpha < 1$) gives strong evidence that, unlike with *FPC-k*, no constant approximation for this problem exists (as, to date, no constant-approximation algorithm for DENSEST-SUBGRAPH was found). We present a variant of the greedy algorithm presented in §2.2 for this problem which fares well in practice.

2.3.1 *GreedyPairsPercentage Algorithm*

We slightly change *GreedyPairs* as follows. *GreedyPairsPercentage*, adds a fifth step to each iteration of *GreedyPairs* to account for clients that have already been α-covered. *GreedyPairsPercentage*'s steps are summarized in Alg. 2:

Algorithm 2 GreedyPairsPercentage

- Ranking: rank each location based on the number of paths in OP traversing it
- Greedy Choice: pick element x with highest rank (breaking ties arbitrarily).
- Input Update : update outstanding paths by removing from P all paths P_x containing element x.
- **Client Update: remove all paths $p \in P^i$ from OP if α paths in P^ihave been covered**
- Termination: if $|R| = k$, stop. Otherwise , repeat from Step 1

As stated in the previous section, the solution R produced by *GreedyPairsPercentage* does not have provable approximation guarantees. In other words, we do not know what kind of relation it has to the ideal optimal solution. Yet, it is of empirical value and a vehicle for comparison against *GreedyPairs*.

3. DRP AGAINST SMART FILTERING

The Internet is a dynamic space and BGP [15], the inter-domain routing protocol, allows an AS to change paths for an IP prefix over time. Without discussing the intricacies of inter-domain routing, we note that an AS X can obtain multiple paths to the same prefix. These *valid* paths are distributed to X by its neighbors.

In terms of the decoy routing scheme, a client could have more than one path to a decoy destination. Having more than one path to the same decoy destination gives a filtering entity, e.g. the ISP, a chance to bypass decoy routers.

Let's assume that the filtering entity can discover the presence of a DR on one of its paths.[4] Once the DR-covered path is discovered, the filterer can choose a DR-free path to the same decoy destination to avoid the DR deployment. An example scenario is shown in Figure 2.

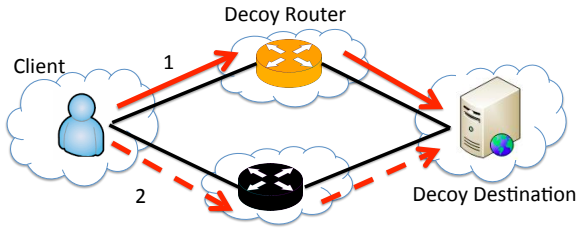


Figure 2: A DR deployment covers the path between a client and a decoy destination (1). A *smart* filtering entity discovers the DR and chooses a neighboring path to reach the decoy destination, thereby avoiding the DR (2).

We refer to this new problem as *decoy router placement against smart filtering* (DRPSF). In DRPSF, an effective candidate solution R covers every valid path between a $< c, d >$ pair. In the following sections we formally define DRPSF, analyze its complexity and propose an algorithm to calculate candidate solutions sets.

3.1 DRPSF Problem Formulation

To analyze DRPSF, we need to introduce the notation for AS neighbors. We denote the set of AS_x's neighbors in the AS graph as N_x. As stated previously, neighbors are important because they provide filtering entities opportunities to avoid the decoy routing scheme.

The introduction of multiple valid paths between a client and a decoy destination identifies a client-destination pair $< c, d >$ with a set of paths P_j^i rather than a single

[4]For example, by obtaining a copy of the decoy routing software and acting as a client.

path p_{ij} (§2.1). Formally, P_j^i is composed of the path p_{ij} plus all feasible paths from N_i to d_j:

$$P_j^i = \{p_{x,j} | x \in i \cup N_i\}$$

The notation is summarized in Table 1.

Overall DRPSF has introduced two changes: the number of paths to cover in P has increased due to neighboring paths and each $< c, d >$ pair now has more than one path. Nevertheless, the metrics to evaluate DRP, (§ 2.1) are still valid for DRPSF.

3.2 Complexity

From a complexity standpoint, losing the single path-per-pair assumption means the approximation guarantees for problem FPC-k no longer hold. Thus, *Greedy-Pairs* no longer provides a constant approximation to the optimal solution with respect to *FPC-k* (§ 2.2).

Posed from a different perspective, though, both the DRP and DRPSF problems relate to an interesting theoretical bound. We refer to *min-DRs* as the problem of finding the *minimum* number of DRs to cover *every path* between clients and decoy destinations. Unlike $FPC - k$, *min-DRs* does not measure partial coverage of pairs, but rather the number of resources used to achieve total coverage. With respect to the *min-DRs* objective, DRP and DRPSF are similar from a theoretical perspective.

We note that the MDCP problem (minimum deployment cost problem without sampling) in [10], in which the objective is to minimize the placement cost of monitors given a monitoring reward requirement, bears a strong resemblance to *min-DRs*. Suh et al. assign a constant reward to each flow monitored. In our case, each path between a client and decoy destination has equal reward value.

min-DRs can be reduced to the MINIMUM-SET-COVER problem in an approximation-preserving manner. MINIMUM-SET-COVER is NP-hard to approximate within a ratio of $\ln(N)$, where N is the universe of elements to be covered. In the case of the decoy routing scheme, the elements to be covered are the paths in P. Thus, *min-DRs* has an approximation bound of $ln(|P|)$ to the optimal solution.

By slightly changing *GreedyPairs* it is possible to achieve a (tight) $\ln(|P|)$ approximation ratio.

3.3 GreedyPairs for Min-DRs Algorithm

To solve for *min-DRs* the only alteration we have to apply to *GreedyPairs* is in its termination condition. Specifically, *GreedyPairs* does not terminate after k iterations, but only after OP is empty.

See explanation of the $\ln(|P|)$ approximation guarantee in [11].

Algorithm 3 GreedyPairs for Min-DRs

- Ranking: rank each location based on the number of paths in OP traversing it
- Greedy Choice: pick element x with highest rank (breaking ties arbitrarily). Add x to R
- Input Update : update outstanding paths by removing from OP all paths P_x containing element x.
- **Termination: if OP is empty, stop. Otherwise , repeat from Step 1**

4. EVALUATION FRAMEWORK

In this section we describe the framework used to support the analysis of the DRP problem. We start by presenting the data used for the analysis in §4.1. All the datasets are publicly available at CAIDA [16]. Then, in §4.2 we discuss the step that precedes the analysis: how we generate the paths P that are the input to a DRP problem. Last, in §4.3 we discuss the core of the analysis: the execution of the algorithms on the paths generated in the previous step.

4.1 AS-Level Topology and Routing Policies

We use CAIDA's dataset on AS relationships [17] to construct the AS network graph. The dataset is a weekly updated archive of *business relationships* between ASes in the network. Relationships are important because they dictate the flow of traffic on the Internet: an AS establishes its *routing policies* based on relationships with its neighbors. Thus, a path between any pair of ASes in the network derives from a valid composition of these relationships.

Unfortunately these relationships are not publicly available. To generate such a dataset, CAIDA collects data[5] from geographically and topologically diverse locations and applies a relationship-inference algorithm to generate and cross-validate the relationships[19].The datasets used for the analysis contain over 150000 relationships to model approximately 50000 distinct ASes.

We also use CAIDA's dataset on AS information to match each AS to its country [20] so we can evaluate DR placement for clients and decoy routers in various countries.

4.2 AS-Path Generation

To evaluate a DRP algorithm, we first need to generate the paths P between clients and decoy destinations. We use the *routing tree algorithm* specified in Goldberg et al. ([21]) to generate such paths.

The algorithm models the flow of traffic by evaluating

[5]BGP table snapshots from Routeviews [18] servers

relationships between neighboring ASes. Starting from a root AS, the decoy destination in our case, relationships with neighbors are considered in a specific order to maintain network stability conditions [22]. First the algorithm takes into account customer-to-provider relationships. A customer AS conveys traffic to its provider AS for any destination. Second the algorithm adds its peer-to-peer links: links between ASes that transit each others traffic to their respective customers. The last stage of the algorithm adds provider-to-customer links. At completion, the tree of links represents paths between every AS in the network and the root AS. We query the tree for paths between clients (nodes in the tree) and the decoy destination (the root of the tree) and add these to P. To obtain all paths between all clients and decoy destinations, we generate a tree for each decoy destination.

Noteworthy is that this process only generates paths from clients to decoy destinations. Where both directions are required (to study Telex in §5.4), the path from the decoy destination to the client is generated by running the routing-tree algorithm with clients as the root of the tree.

Overall, a set P can range in size from a few hundred paths to several million paths. The size depends on the sizes of sets C and D as well as on the amount of information present in the original relationship dataset.

4.3 DRP Algorithm Execution

The core part of the analysis is the execution of the algorithms proposed in § 2.2.1 and § 2.3.1 on the set P generated in the previous step. The algorithms start by scanning the files containing P and storing their states: information on paths, pairs and clients. The appropriate data structures are built to facilitate the successive iterations of the algorithm; in particular, mappings between nodes/edges and pairs, pairs and clients, as well as rankings for each node/edge. Once the entire set has been parsed, the algorithms iterate to find the candidate solution. In each iteration a candidate node/edge location is chosen by scanning the rankings. The mappings are updated accordingly.

Running the algorithms on very large data sets (10-15 GB) requires large amounts of memory. To meet those memory requirements we used cloud services. We run the analysis on the extra large VM instances of Amazon EC2 that have up to 34 GBs of RAM memory.

5. RESULTS

We define a DRP problem as a function of six variables *DRP(C,D,P,M,F,S)*: clients, decoy destinations, paths, metrics, filtering levels and solution types.

Clients C : We associate clients to countries. In other words, a set of clients is the set of all ASes in a specific country. The matching is based on CAIDA's AS information dataset [20]. Associating a set of clients with a country is appropriate because traffic filtering is often dictated by government entities. We examine eight countries of various sizes and geo-locations and label them (at random) as countries A-H.

Decoy Destinations D : We choose three decoy destination sets of different sizes and properties: ROW, U.S.A. and E-commerce. ROW (*rest-of-the-world*) considers every AS outside the client set C as a decoy destination. This is the ideal set for any DR scheme because everything is a potential decoy. U.S.A. is the decoy destination set of all ASes in the United States. It represents a large fraction of popular destinations on the Internet. E-commerce represents a small set of popular web commerce sites. Decoy Routing and Telex [8, 9] leverage the TLS protocol, used in e-commerce transactions.

Paths P : We include different path properties to accommodate for different implementations of the Decoy Routing architecture. Decoy Routing [8] and Cirripede [7] only need DRs to be placed on the forward path between the client and decoy destination. Thus, we only need to consider paths in one direction (*unidirectional* paths). Telex, instead, needs a DR to be present on both the forward and return path. We defined this the *bidirectional* paths requirement in §2.1.

Metrics M : We have proposed two algorithms to optimize the metrics we have defined in §2.2 and §2.3. We evaluate the solution of a problem by applying a specific algorithm and measuring the "goodness" of the solution proposed through its respective metric.

Filtering Level F : In §3 we have labeled a filtering entity's attempt to work around decoy routing as **smart** filtering and defined the DRPSF problem. We refer to the original DRP problem, where no attempt to work around decoy routing is made by the filtering entity, as *naïve* filtering.

Solution Type S : We consider two strategies for picking candidate solutions: one focuses on nodes in the AS-level graph, the other on edges. Variables and their values are summarized in Table 2.

Clients C	9 Countries
Decoy Destinations D	E-commerce U.S.A. ROW
Paths P	Unidirectional Bidirectional
Metrics M	FPC FαCC
Filtering F	Naïve Smart
Solution Type S	Nodes Edges

Table 2: Decoy Router Placement Problem Variables

In the following sections we analyze the DRP problem by focusing on each variable. The approach we follow is to define a specific **scenario** as a combination of the six DRP variables and to compare results when one or two variables change. The goal is to receive a wider spectrum of results from which to draw conclusions and to

highlight the importance of each variable in the problem. We then tune multiple parameters at once to get a better understanding of how many decoy routers might be required in extreme scenarios.

5.1 Decoy Routers on Nodes vs. Edges

We start by comparing solutions comprised of nodes or edges. Picking nodes provides a coarser analysis: for each picked node we assume we can intercept all traffic traversing the AS.[6] Picking edges provides a more fine grained analysis as we restrict the location of the DR down to inter-AS links. We study the two solution types across every country from which we have collected data and set the decoy destinations set to be the largest possible (ROW). We consider only paths in the forward direction (unidirectional) and a naïve filtering level. We apply the *GreedyPairs* algorithm and evaluate the solution by metric FPC(§ 2.2). The results are shown in Figure 5.1. The country names are anonymized. The two figures plot the Complementary Cumulative Distribution Function (CCDF) of the fraction of pairs covered. In other words, the fraction of pairs still to be covered given a number of nodes or edges.

Both graphs show curves with similar behavior: an exponential decrease followed by an asymptote. The exponential decrease means a small set of elements (either nodes or edges) cover the majority of pairs. Adding these popular locations to the solution is of tremendous value. After a given number of elements, the marginal value of adding more becomes negligible (asymptote). This break point can be considered as the optimal trade-off between pair coverage and resources used.

Country	Nodes	Edges
A	0.89	0.47
B	0.92	0.74
C	0.97	0.64
D	0.95	0.84
E	0.99	0.99
F	1.0	0.97
G	1.0	0.99
H	1.0	1.0

Table 3: Fraction of $< c, d >$ pairs covered with 30 nodes/edges

Table 3 shows the fraction of pairs covered with a fixed number of elements k. The value k is picked to the right of every break point in each curve. The fraction of paths covered with a fixed number of edges is lower with respect to nodes.

Placing a DR on one inter-AS link is not the same as placing DRs on an entire AS, yet concentrating DRs on a small number of specific locations may prove easier than distributing DRs across distant and disparate locations. Furthermore, even for larger countries that present a larger number of ASes (e.g., countries A, B, and C) concentrating DRs on a few nodes maintains a high coverage value. Thus, we conclude the following: 1) 'popular' nodes receive decoy routing traffic from several links 2) focusing on a small number of specific locations achieves high coverage no matter the geo-location of the client set. We suspect high coverage is achieved regardless of the client set because 'popular' nodes are large ASes, probably Tier-1 ASes. We investigate this matter further in following sections (§ 5.6).

From this point forward we will continue our analysis by focusing on whole-node placement instead of edge placement.

5.2 Locations of Decoy Destinations

To make filtering hard we would like for every AS outside of the client set to be a potential decoy destination. Nevertheless, choosing a small set of decoys could scale the deployment of DRs down and still prove to be difficult for the filtering entity to block. Figure 4 shows the CCDF across different decoy destination sets. Each data point in the curves represents the median value over all countries from which we collected data. The horizontal red line marks 90% coverage.

The other scenario variables (paths, metrics, filtering) are equal to the ones used in the previous section.

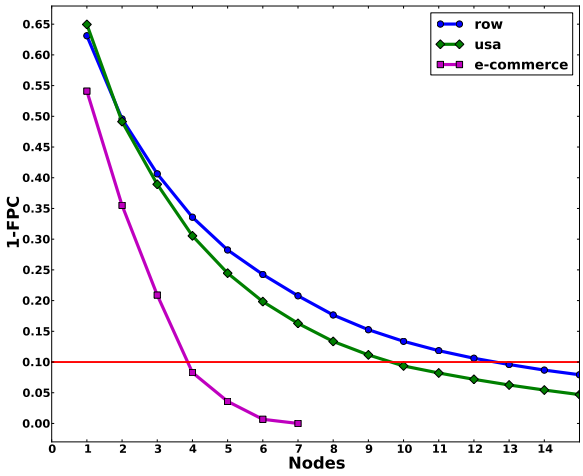
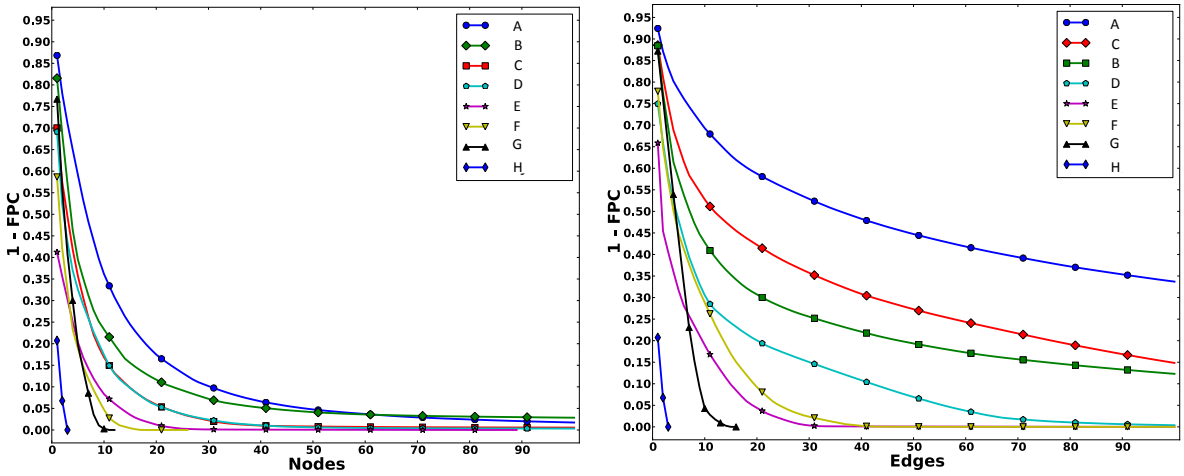


Figure 4: Comparison between GreedyPairs solutions for different decoy destinations sets. The red solid line is for 90% coverage

Our results show an almost insignificant growth in the number of nodes needed to be deployed for very differently sized decoy destination sets. The E-commerce set is comprised of four ASes and needs a comparable number of nodes to the decoy destination set to be covered. Note that our algorithm is an approximation and did not produce the optimal result, the four e-commerce sites themselves. The larger U.S.A. and ROW sets have more than 15,000 destination ASes to cover and only

[6]This is achieved by instrumenting all of its ingress-egress routers.



(a) (b)

Figure 3: Nodes and Edges analysis across different countries. The x axis represents the number of elements, the y axis the CCDF of fraction of pairs covered.

need ten and thirteen DRs respectively to achieve the same fraction of pairs covered. This shows that it is desirable to consider large decoy destination sets for real DR deployments.

5.3 Naïve vs. Smart Filtering

In §3 we introduced the DRPSF problem as the variation of the DRP problem when facing a proactive, smart filtering entity. In this part of the analysis we assess the differences between solutions proposed when applying the *GreedyPairs* algorithm against a naïve and smart adversary. It is worth remembering that the main difference between DRP and DRPSF is that paths P in DRP are augmented by valid neighboring paths in DRPSF's P'. Furthermore, a $< c, d >$ pair is now described by more than one path in the DRPSF problem.

The scenario considered is the following: the decoy destination set is ROW, DRs are unidirectional, and we applied *GreedyPairs* to P and P'. We evaluated the solutions through metric FPC and the resulting CCDFs are in 5. Each curve in the plot represents the median value across all sampled countries.

The smart filtering curve presents the same behavior as the naive one: an exponential decrease followed by an asymptote. Thus, even with a higher level of filtering, picking a small number of popular locations accounts for coverage of the majority of pairs.

The difference between the two curves is that the smart filtering one presents lower coverage values with respect to the naive curve. This difference is roughly 10%.

Our interpretation is that this is because most of the paths in P' that are not present in P (e.g., paths from clients' neighbors to the decoy destinations) intersect at locations where traffic from different ASes often converges (e.g., in Tier-1 ASes).

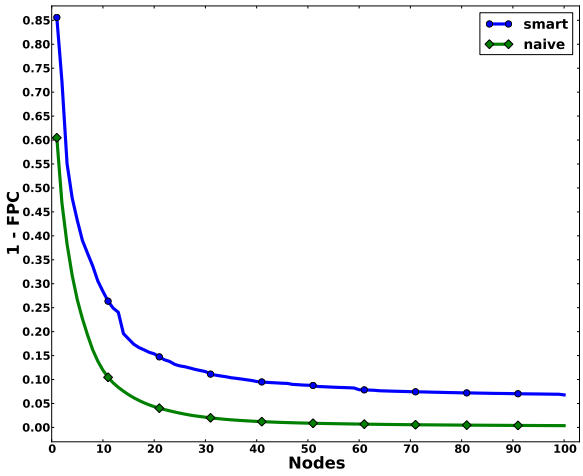


Figure 5: Comparison between naïve (DRP) and smart filtering (DRPSF) problem solutions proposed by algorithm GreedyPairs.

5.4 Unidirectional vs. Bidirectional Paths

In § 2.1 we described the different requirements the implementations of Decoy Routing, Cirripede and Telex [8, 7, 9] have in terms of DR placement. Decoy Routing and Cirripede only need the DR to lie on the forward path between a client and decoy destination. Telex requires the DR to lie on both the forward and return path. We defined this difference as the *unidirectional* and *bidirectional* paths requirements. In the bidirectional paths case, the forward and return path might not coincide (paths are not guaranteed to be symmetric on the Internet). Therefore, we calculate paths from C to D as well as D to C and for each $< c, d >$ we can only nodes or edges present in both paths are candidate DR locations.

We consider a scenario similar to the ones in the previous sections: decoy destination set is ROW and filtering is naïve.

Figure 6 shows the CCDF of metric FPC metric for bidirectional and unidirectional paths. Each curve is the median values across all countries.

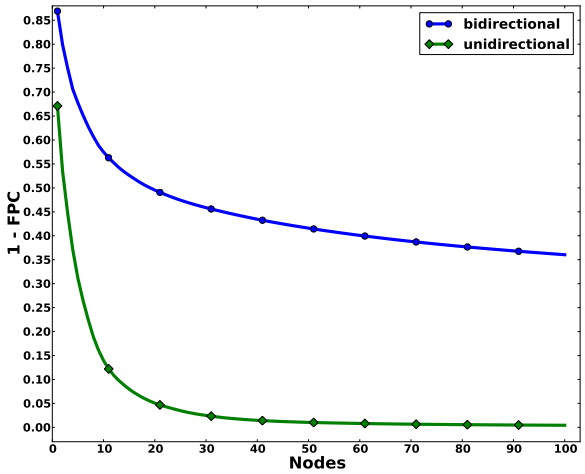


Figure 6: Unidirectional - Bidirectional Paths Comparison

The bidirectional paths curve shows lower values of pair coverage (higher CCDF values) with respect to the unidirectional paths curve. The asymmetry of Internet routing makes it hard for forward and return paths to intersect anywhere but at the decoy destination. This property of the network offers advantages to the Cirripede and Decoy Routing implementation which only need to lie on the forward direction. It seems that a Telex deployment would prove to be less efficient and largely restrict the number of candidate locations.

Nevertheless, the bidirectional curve is similar to those we have seen in previous analyses: a small set of popular locations is sufficient to cover the majority of pairs. We reiterate our belief that such locations are found in the backbone of the network and associate them with Tier-1 ASes.

5.5 Coverage Metrics

So far in our analysis we have only applied the algorithm *GreedyPairs* and evaluated solutions by metric FPC. In § 2.3 we proposed a metric centered around client-coverage, FαCC, and an algorithm, *GreedyPairsPercentage*, to optimize solutions for this metric. *GreedyPairsPercentage* does not have the approximation guarantees of *GreedyPairs*, yet it might prove worthy of finding more efficient DRP problem solutions. In this part of the analysis we will compare the two algorithms and evaluate solutions through metric FαCC (§2.3).

We consider two very similar scenarios. In both, the set of decoy destinations considered is ROW and the filtering level is smart. The first scenario, though, accounts for unidirectional paths while the second assumes bidirectional paths. The results for both scenarios are shown in Figure 7.

In the figure, *GreedyPairs* is labeled as 'gp' and *GreedyPairsPercentage* as 'gpp'. The solid lines are the curves for the unidirectional paths scenario. As we can see, there is not a significant difference between them. Thus, *GreedyPairs* does well at α-covering clients. The interpretation is that paths from a client to all decoy destinations, represented by set P^i, are similar among each other. In conjunction with the results from § 5.3, we conclude that a client does not only reach one decoy destination through similar paths, but reaches also different decoy destinations through similar paths.

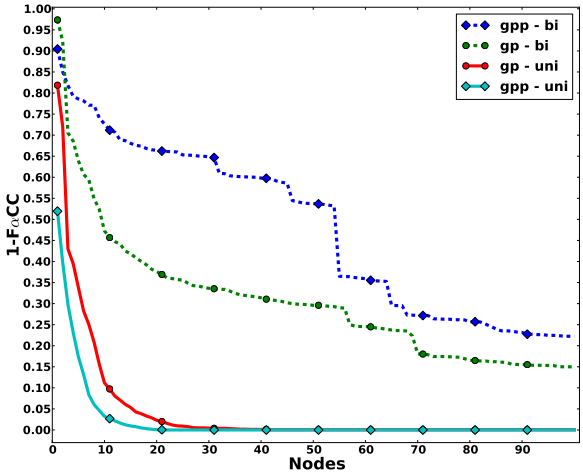


Figure 7: Comparison between algorithms GreedyPairs - GreedyPairsPercentage over metric FαCC.

The dashed curves refer to the scenario with bidirectional paths. These two curves show, surprisingly, a considerable difference between *GreedyPairs* and *GreedyPairsPercentage*. In fact, *GreedyPairs* outperforms *GreedyPairsPercentage*. The requirement of DR deployment at the intersection of forward and return paths forces the algorithms to pick 'popular' locations. Picking 'popular' locations is the strategy followed by *GreedyPairs*. *GreedyPairsPercentage*, instead, preemptively excludes paths if a client is α-covered. This exclusion, we believe, decreases the popularity of certain locations and induces the greedy choice to be less effective in later stages of the algorithm.

In conclusion, we believe the popularity of locations is heavily skewed in the network. This distinguishes a set of locations that are traversed by the majority of paths, i.e. Tier-1 ASes, from others which are traversed by a small number of paths, i.e. stub ASes. If we artificially take the skewness away, we inadvertently also exclude popular locations from our solution.

5.6 Decoy Routers Locations

We have given a wide range of perspectives on the DRP problem by analyzing each of its variables and how these effect the solutions proposed by algorithms *GreedyPairs* and *GreedyPairsPercentage*. In the interpretation of the results, we have shown that the DRP

problem can be efficiently solved with a small set of decoy routers placed at very popular locations. We have hypothesized that such popular locations are large ASes, i.e. Tier-1 ASes.

To confirm such statements, we match the locations proposed in the candidate solutions against three specific AS sets. The first AS set contains 19 ASes confirmed to be Tier-1 ASes [23]. The other two sets are, respectively, the top 100 ASes with highest out-degree and the top 100 ASes with largest customer cone[7] [24].

We take the following approach: we focus on node analysis, fix the decoy destination set to ROW and we explore the candidate solutions of all countries with different path / filtering variable values. For each combination of variables we intersect the candidate solution set produced by the GreedyPairs algorithm with the afore-mentioned AS sets.

Table 4 shows the results of the intersection between the candidate solutions and the three AS sets. Each row is a combination of the path/filtering variables and represents the average across all countries of the fraction of ASes in the candidate solution sets belonging to the three large AS sets.

DRP Variables	Large AS Sets		
	Tier 1	Top 100 Out-Degree	Top 100 Customer Cone
naïve - uni	0.5	0.3	0.47
naïve - bi	0.98	0.95	0.95
smart - uni	0.59	0.4	0.55
smart - bi	0.86	0.45	0.54

Table 4: Fraction of candidate solution elements belonging to large AS sets, averaged across all countries and for different path and filtering variables.

The results show that sets of large ASes are present in candidate solutions in every scenario. In fact, at least half of the locations chosen for any country are taken from the Tier-1 AS set. This number increases when we consider bidirectional paths. In particular, the naïve filtering-bidirectional paths scenario indicates that across all countries the solution set is composed primarily of Tier-1 ASes. The fraction of ASes from the top 100 out-degree and customer cone sets are also significant: when not Tier-1 ASes, candidate locations are large ASes.

We conduct a further analysis to relate the locations indicate in the solutions with their relative position in the paths between clients and decoy destinations. The goal is to understand where in the paths DRs are mostly situated to gain insight about the structure of the network graph. We distinguish between two positions on each path: close to the endpoints, e.g. client and decoy destination, or in the 'middle' of the path. We associate the middle of the path with positions 'deeper' in the network, where we assume to find Tier-1 and Tier-2 ASes.

We use the scenarios proposed earlier in this section: node analysis, all countries, ROW decoy destination set and different path properties and filtering level. Figure

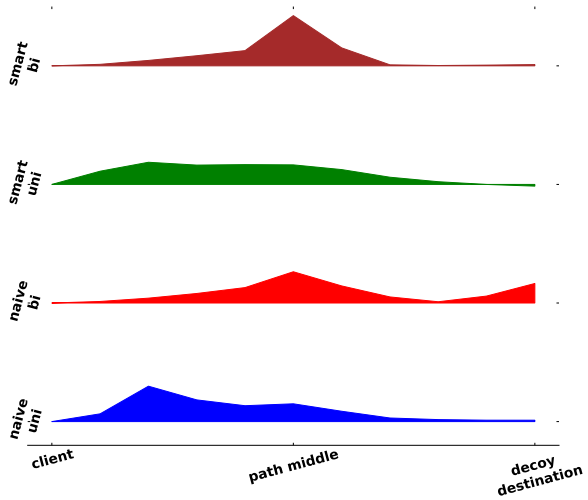


Figure 8: Average Probability Distribution Function of the relative position of candidate DRs in client-decoy destination paths for different combination of path and filtering variables.

8 shows the results of our analysis. Again we distinguish between combinations of the variables for path properties (unidirectional, bidirectional) and filtering level (naïve,smart). For each combination, we show the probability distribution function of the locations designed in the solution. The distribution is the median distribution across all countries.

We analyze each distribution, starting from the bottom of the figure. In the naïve-unidirectional (naïve-uni) case the distribution leans more towards the client: we believe it is because the clients set presents a number of 'smaller' countries. For these smaller countries the best locations are closer to their borders. In the naïve-bidirectional (naïve-bi) case the majority of DRs are located in the middle of the path or near the decoy destinations. As we have discussed in § 5.4 forward and return paths may not present any intersections apart from the decoy destination. These results show indeed that many intersections only happen at the decoy destination but there is also a significant number of intersections deeper in the network. The smart-unidirectional (smart-uni) case presents a similar shape to the naïve-unidirectional one. The difference is a distribution more skewed towards the middle of the paths. The last distribution combines smart filtering and bidirectional paths. If we intersect the distributions from the previous two cases, the great majority of locations for DRs is deeper in the network.

In summary, we can relate the results from Table 4

[7]AS Y is in the customer cone of AS X if Y is directly or indirectly a customer of X

and Figure 8: we have discovered that the small set of popular locations that characterized the results in each of the previous sections is indeed a set of large, possibly Tier-1, ASes. Due to the hierarchical structure of the network graph, these large ASes are found in the middle of paths, equidistant from both clients and decoy destinations.

5.7 Country-Independent Deployment

So far we have focused on sets of clients that represent a single country. Ideally, though, decoy routing is to be made available to any user, regardless of her nationality. The overlap of different solutions with small sets of large ASes, as seen in Section 5.6, makes us believe that there are strong similarities across country deployments. Such similarities would make it possible to have one deployment that is country-independent and satisfies many users worldwide.

To confirm similarities across sets of countries we match a country's candidate solution with a set of paths from another country and record the fraction of pairs (FPC) covered. We repeat this process for different scenarios. In our analysis, we take the solutions proposed by GreedyPairs for one country, which we will call country X, and match them against the data sets of the other countries. We analyzed several scenarios: we fixed the variables for the decoy destination set (ROW) and metric (FPC) and considered all possible combinations for the others (solution elements, paths, and filtering).

Results show that there is a high overlap of candidate solutions across countries. In particular, this is true for solutions comprised of nodes: the FPC by X's deployment across all data sets is within 10% of the FPC by the original country. For edge-based solutions the coverage decreases dramatically. Variables for path properties and filtering level have similar behavior to previous analyses: placing a DR at the intersection of forward and return paths (bidirectional paths) generates difficulties in coverage across countries, while moving from naïve filtering to smart filtering only decreases coverage by a small amount (5%).

Overall we believe that a country independent deployment which satisfies a large number of users is possible: the structure of the network graph makes Tier-1 ASes the ideal location for DR deployment for any set of clients.

6. RELATED WORK

The problem of DR placement in the network is similar to the well known and deeply studied problem of *monitor* placement. Monitor placement is categorized into active and passive monitor placement.

Active monitor placement focuses on the deployment of devices capable of probing the network to infer its topological properties. In this space, Jamin et al. [25] propose the IDMaps project to calculate relative distances between hosts in the network. Their scope is different from ours, but their results on dimishing returns are similar to ours with DRs. We also highlighted a 'break point' after which there is very small value in adding more decoy routers.

Horton et al. [26] focus on the problem of deploying active beacons to monitor the connectivity in the network. Their goal is to deploy the minimum number of beacons to infer the status of every edge in the network. This problem is equivalent to the *min-DRs* problem in § 3.2. To solve it, Horton et al. use a greedy heuristic based on the max connectivity of a node called *arity*. We draw similarities to Horton's work for their intent to correlate the beacons problem to the structure of the real network. In fact, Horton et al. claim that aiming for nodes with higher connectivity is an effective strategy to optimizing beacon placement. Similarly, we claim that DR deployment is most valuable in big ASes deeper in the network.

Passive monitor placement focuses on maximizing traffic monitoring while minimizing deployment of monitors. Suh et al. [10] focus on two monitor placement problems similar to $FPC-k$ §2.1 and $min-DRs$ §3.2: Budget Constrained Maximum Coverage and Minimum Deployment Cost. In the former, the goal is to maximize the number of flows sampled with a fixed amount of monitors. The latter is the dual problem: minimizing the number of monitors deployed to sample a given number of flows. Flows can be compared to paths in the decoy routing model. We differ from Suh's model as we focus on pairs and clients rather than flows. Furthermore, Suh et al. evaluate their model on smaller, synthetically generated networks while we rely on inferred data that spans across the entire network.

Jackson et al. [27] assess the problem of complete network monitorage. The goal of their analysis it to monitor every valid path between nodes of the AS-level network graph. Jackson et al. rank AS popularity by topological information of the network (out-degree of ASes involved). Then they propose to follow two strategies: depth first - instrumenting the N most popular links in the network - and breadth first - instrumenting the most popular inter-AS link of the top N ASes. Their results show that a breadth first deployment is more successful. We focus on instrumenting ASes rather than inter-AS links because we have found that for specific client sets, covering an AS completely yields higher coverage values.

7. CONCLUSION

Our analysis has shown many facets of the DRP problem. We have taken into consideration several client and decoy destination sets, unidirectional and bidirectional paths as well as a variation of the original problem

where filtering entities react to decoy routing.

The results highlight great similarities across solutions to different DRP problem scenarios. In particular, there always exists a small set of 'popular' locations where it is extremely valuable to have decoy routers. Equipping these locations accounts for the vast majority of paths between sets of clients and decoy destinations, no matter their size or geo-location. A look at the position of these location has shown that a DR deployment has to happen 'deep' in the network, in large Tier-1 and Tier-2 ASes. Furthermore, a DR deployment in such ASes provides good coverage not for a single country, but across different countries. Outside of this set, adding more decoy routers yields diminishing returns and full coverage may not be feasible.

The goal of decoy routing, though, is not to achieve total coverage between clients and decoy destinations but to be pervasive enough to make it difficult for a filtering entity to effectively block its users. We have shown that this is achievable even in the case that every possible destination is considered a decoy, even if the DR has to lie on both the forward path and return path from the client to the decoy destination, and even if the filtering entity proactively tries to avoid decoy routing.

Overall, we believe that decoy routing is a valid response to network filtering and that its biggest challenge, the necessity to lie on the path between a client and a destination, can be solved efficiently.

Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Navy SPAWAR under Contract N66001-11-C-4017. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. Distribution Statement "A" (Approved for Public Release, Distribution Unlimited).

8. REFERENCES

[1] ONI, "Opennet initiative." `http://opennet.net/research/profiles`.

[2] BBC News, "Tehran blocks access to Facebook." `http://news.bbc.co.uk/2/hi/8065578.stm`, May 2009.

[3] TOR, "Tor partially blocked in China." Blog post at `https://blog.torproject.org/blog/tor-partially-blocked-china`.

[4] Global Pass, "Global pass." `http://gpass1.com/gpass/`.

[5] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proc. Usenix Security Symposium*, 2004.

[6] R. Deibert, "Psiphon." `http://psiphon.civisec.org/`.

[7] A. Houmansadr, G. T. Nguyen, M. Caesar, and N. Borisov, "Cirripede: Circumvention infrastructure using router redirection with plausible deniability," in *Proc. ACM Conference on Computer and Communications Security*, pp. 187–200, 2011.

[8] J. Karlin, D. Ellard, A. Jackson, C. Jones, G. Lauer, D. Mankins, and W. Strayer, "Decoy routing: Toward unblockable Internet communication," in *Proc. USENIX Workshop on Free and Open Communications on the Internet*, 2011.

[9] E. Wustrow, S. Wolchok, I. Goldberg, and J. A. Halderman, "Telex: Anticensorship in the network infrastructure," in *Proc. USENIX Security Symposium*, August 2011.

[10] K. Suh, Y. Guo, J. Kurose, and D. Towsley, "Locating network monitors: complexity, heuristics, and coverage," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 1, pp. 351 – 361 vol. 1, march 2005.

[11] U. Feige, "A threshold of ln n for approximating set cover," *J. ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[12] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Inf. Process. Lett.*, Apr. 1999.

[13] U. Feige, D. Peleg, and G. Kortsarz, "The dense k-subgraph problem," *Algorithmica*, vol. 29, pp. 410–421, 2001. 10.1007/s004530010050.

[14] D. Goldstein and M. Langberg, "The dense k subgraph problem," *CoRR*, vol. abs/0912.5327, 2009.

[15] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)." RFC 4271 (Proposed Standard), Jan. 2006.

[16] CAIDA, "The Cooperative Association for Internet Data Analysis." `http://www.caida.org`.

[17] CAIDA, "AS relationships dataset." `http://www.caida.org/data/active/as-relationships/`.

[18] RouteViews, "RouteViews Project." `http://www.routeviews.org/`.

[19] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, k. claffy, and G. Riley, "AS relationships: Inference and validation," *ACM SIGCOMM Computer Communication Review (CCR)*, pp. 29–40, January 2007.

[20] CAIDA, "AS information dataset." `http://as-rank.caida.org/data/2011.01/as2info.2010.08.txt`.

[21] S. Goldberg, M. Schapira, P. Hummon, and J. Rexford, "How secure are secure interdomain routing protocols," in *"Proc. ACM SIGCOMM"*, pp. 87–98, aug 2010.

[22] L. Gao and J. Rexford, "Stable Internet routing without global coordination," *IEEE/ACM Trans. Netw.*, vol. 9, pp. 681–692, Dec. 2001.

[23] Wikipedia, "Tier-1 network." `http://en.wikipedia.org/wiki/Tier_1_network`.

[24] CAIDA, "AS ranking." `http://as-rank.caida.org/`.

[25] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "On the placement of internet instrumentation," in *In IEEE INFOCOM 2000, Tel Aviv*, pp. 295–304, 2000.

[26] J. D. Horton and A. López-Ortiz, "On the number of distributed measurement points for network tomography," in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, IMC '03, (New York, NY, USA), pp. 204–209, ACM, 2003.

[27] A. Jackson, W. Milliken, C. Santivanez, M. Condell, and W. Strayer, "A topological analysis of monitor placement," in *Network Computing and Applications, 2007. NCA 2007. Sixth IEEE International Symposium on*, pp. 169 –178, july 2007.