# Proximax: A Measurement Based System for Proxy Dissemination

Damon McCoy[1] and Jose Andre Morales[2] and Kirill Levchenko[1]

[1] University of California at San Diego
{dlmccoy,klevchen}@cs.ucsd.edu
[2] Institute for Cyber Security, University of Texas at San Antonio
jose.morales@utsa.edu

**Abstract.** Many people currently use proxies to circumvent government censorship that blocks access to content on the Internet. Unfortunately, the dissemination channels used to distribute proxy server locations are increasingly being monitored to discover and quickly block these proxies. This has given rise to a large number of *ad hoc* dissemination channels that leverage trust networks to reach legitimate users and at the same time prevent proxy server addresses from falling into the hands of censors. To address this problem in a more principled manner, we present *Proximax*, a robust system that continuously distributes pools of proxies to a large number of channels. The key research challenge in *Proximax* is to distribute the proxies among the different channels in a way that maximizes the usage of these proxies while minimizing the risk of having them blocked. This is challenging because of two conflicting goals: widely disseminating the location of the proxies to fully utilize their capacity and preventing (or at least delaying) their discovery by censors. We present a practical system that lays out a design and analytical model that balances these factors.

## 1 Introduction

Internet censorship is a fact of life for most of the world's population. While the stated purpose of such censorship is to protect users from harmful content, political news and commentary are often blocked as well. Reports of government censorship in the wake of anti-government protests in Burma and Iran [1–4] underscore the growing role of the Internet in enabling political speech and organization, as well as the steps taken by governments to control it.

To circumvent censorship, users rely on external Internet *proxies* [5,6]. In the simplest case, this is simply a SOCKS proxy supporting encrypted connections (e.g. TLS). Encrypting the connection to the proxy provides content *confidentiality*, thus bypassing content filtering.[3] Widespread use of proxies has in turn led to *secondary* censorship: governments identifying and blocking the proxies themselves [7–9] (at the network level).

[3] In addition to confidentiality, proxies also provide *anonymity* by aggregating all users behind a singe host.

Individuals and organizations providing proxy service are thus faced with the additional challenge of advertising their resources to their target audience while preventing the same information from falling into the hands of censorship authorities. The Tor network [10], for example, has recently added *bridge relays* that offer an improved level of censorship resistance by relay traffic to publicly-advertised Tor core routers (which have become blocked by censors) [11]. In response, the Chinese government has enumerated and blocked all Tor bridge relays advertised via the website distribution channel [9].

Today, addresses of open proxies are distributed via *ad hoc* mailing lists or via social networking sites such as Facebook and Twitter. Such "trust networks" provide a degree of protection against discovery by censorship authorities; however they also limit the population served by these proxies. Negotiating this trade-off between publicity and secrecy is no easy task: advertising to more people means greater effectiveness, but also greater risk being blocked.

Our proposed solution is to cast the problem as that of maximizing *yield*, that is, the number of user-hours of service provided by a set of proxies. Proxy addresses are given to a set of *registered users* to advertise in any manner they wish. *Proximax* provides a means of estimating each user's effectiveness, and a policy for choosing the most effective users for advertising proxies, with respect to our objective of maximizing total system yield.

The contributions of this paper are as follows:

- We cast the proxy advertisement problem as one of choosing the most effective set of registered users the job of advertising new proxies should be delegated, with the objective of maximizing the total user-hours of service provided by the system.
- We describe a system for estimating the effectiveness of each user with respect to this objective, and show how to choose the best set of users for advertising new proxies.

## 2 Design

*Proximax* is a proxy distribution system in which users themselves are the means of disseminating proxy addresses. Users disseminate proxy addresses to their friends in any manner they wish, whether via a private mailing list, social networking site, or in person. Their friends, in turn, pass the information on to their friends, and so on.

There is a special set of users—*registered users*—who learn proxy addresses directly from the system; all other users learn about proxies from other users. *Proximax* is responsible for determining the effectiveness of each registered user,

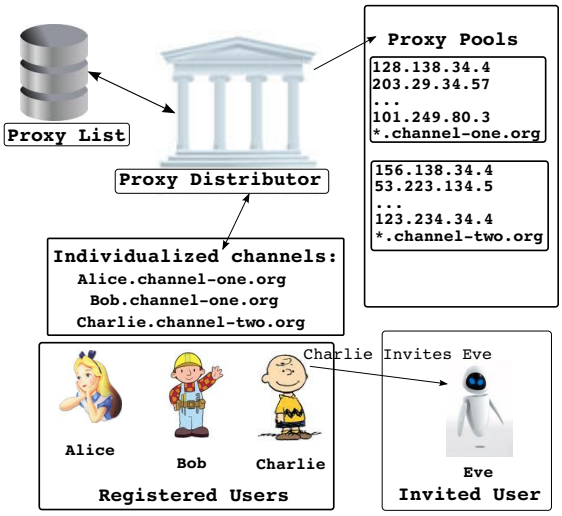


**Fig. 1.** *Proximax* system components.

### 2.1 Challenges

Anti-censorship systems typically distribute lists of proxies to end users through informal social networks such as message boards, Twitter, instant messages, and mailing lists [6,12,13]. In contrast to previously proposed open proxy distribution systems [14, 15], *Proximax* tracks both the usage rate and the risk of proxies distributed via different channels being blocked. The goal of *Proximax* is to maximize the useful service of the limited number of proxies that we have to disseminate. It is important to note that the amount of useful service of a proxy depends both on how many end users it attracts and on how long the proxy remains in operation (before being blocked). We measure of useful service as the number of user-hours a proxy provides, which we refer to as its *yield*. Our objective is to maximize total system yield given a limited number of proxies. To do so we must balance two conflicting goals: widely disseminating proxy addresses to fully utilize their capacity and evading discovery by censors.

### 2.2 System Tasks

The operation of *Proximax* consists of three main tasks illustrated in Figure 1. We present an overview of these tasks here.

**Disseminating Proxies.** We assume that there is a trusted group of administrators who run the *Proximax* system. Administrators have a full list of proxies. Each registered user receives an individualized host name which they can disseminate to friends. The fact that the host name is individualized allows us to track how many additional end users a channel brings to a set of proxies, which affects their standing in the system (more on that later). We envision that a channel could be a private email list, social networking site, or a censorship-resistant

publishing tool, such as Collage [16]. These addresses may be discovered by the adversary and blocked either by infiltrating the distribution chain or some other method. Our system is built on the assumption that this will eventually happen with all addresses. We also assume that *Proximax* can collect statistics on the usage of these proxies by country to determine if they should be removed from the system when they are either blocked or fail.

**Managing Channels.** As previously stated, each registered user has an individualized host name (which take the form of a unique domain name registered with DNS). In order to make it difficult to discover and ban channels we piggyback on the DNS infrastructure, using a technique, commonly employed by botnets and malware distributors, called *fast flux* [17]. As part of this technique *Proximax* will register multiple proxies to the same domain name and uses round-robin DNS along with short Time-To-Live (TTL) values to create a constantly changing list of proxies for that single domain name. This additionally allows *Proximax* to automatically load balance resources by adding and removing proxies based on current utilization levels.

The adversary can block the channel (DNS blocking) at which point *Proximax* will issue another individualized host name to that channel. The adversary can also block all of the individual proxy addresses (IP level blocking) assigned to that channel when it was discovered[4]. If they block all the individual proxies but not the channel *Proximax* might still want to provide a new host name to the channel to force the adversary to discover the new name.

**Inviting Users.** There must be some mechanism within *Proximax* to allow the registration of new users. We feel that an open registration process would allow the adversary to flood the system with fictitious registered users. Thus, we choose to only allow new users to enter the system through an invitation by an existing registered user. These invites should be limited and the number of new invitations will be based on the current under-utilization of the system (if any) to increase the usage rate of resources. Which potential new registered users are granted invitations will be based on their pre-existing performance, if any, as a non-registered user and historical performance of the registered user issuing the invitation. The performance of current registered users invited by the same inviter is also considered. If the inviting registered user has never issued an invitation, granting a new invitation is based solely on the inviter's direct performance.

Approving new invitations is achievable with a reputation-based analysis of the inviting registered user's subtree. The inviting registered user is the root of a subtree where each subnode is a current registered user which was invited by the root or one of its descendants. The performance of the root and each subnode produces a reputation score for the root based on user-hours and number of blocked resources from the root and all subnodes in the subtree. Analyzing the complete subtree is essential to identify nodes that are not a direct child of the root which have performed poorly. Similar to the RICO Act [18] in the legal system, once a subnode is suspicious the whole subtree is equally suspicious. A low reputation score is a non-trusted root user and the invitation is denied. A

[4] This will affect other channels that are sharing this same resource.

high score is a trusted root user and the invitation is granted. A middle score requires further analysis beyond the scope of this paper.

## 3 Analysis

The preceding section described the overall system for allocating and managing proxies; in this section we fill in the details of how user effectiveness (i.e. attracted user-hours) is estimated, and how this estimate is used to decide which registered users should be chosen to advertise a new proxy.

### 3.1 Model

Abstractly, we model our problem as one of choosing a set of *dissemination channels* (or more simply *channels*) to use to advertise a set of *resources*. In our case, proxies play the role of resources and registered users the role of dissemination channels. Advertising a resource attracts a certain level of resource usage based on the channel used to advertise the resource. Advertising a resource also carries the risk of the resource being discovered and blocked, rendering it unusable.

| | |
|---|---|
| $m$ | Number of resources. |
| $n$ | Number of channels. |
| $\gamma$ | Intrinsic resource risk (parameter). |
| $R_i$ | Set of resources advertised via channel $i$. |
| $t_i$ | Resource $i$ lifetime (measured). |
| $\lambda_j$ | Channel $j$ risk (unknown). |
| $u_j$ | Channel $j$ attracted usage (estimated directly). |
| $\Lambda_i$ | Total resource risk; Eq. (1). |
| $U_i$ | Total resource $i$ usage; Eq. (1). |

**Table 1.** Parameters and notation used in Section 3.

Formally, we have a set of $m$ identical resources and $n$ channels. Each channel has an associated level of *usage*, denoted $u_j$. In our system, we measure usage as the number of user-hours the channel attracts per day. We assume that the usage level is stable or changes only slowly over time, and thus easy to estimate.

Each channel also contributes a level of risk to the resource. We model this risk as a Poisson process; that is, we assume that during each infinitesimal period of time there is a fixed probability of the resource being blocked or shut down. This risk is quantified by the Poisson process rate parameter $\lambda_j$, where the probability of a resource being shut down at or before time $t$ as a result of being advertised on channel $j$ is $1 - e^{-\lambda_j t}$. If the Poisson processes associated with each channel are independent, then the rate parameters are additive: a resource is advertised on two channels $j$ and $j'$ has rate parameter $\lambda_j + \lambda_{j'}$, so that the probability of a resource being shut down at or before time $t$ as a result of being advertised on channel $j$ and channel $j'$ is $1 - e^{-(\lambda_j + \lambda_{j'})t}$. Note that this holds only if each channel is independent with respect to its risk of being censored; while this may not be a realistic assumption, we believe it provides a reasonable first-order approximation.

Because resources are identical and can be advertised immediately (Section 2), there is no benefit to advertising more than one available resource on a channel.[5] We also assume that resources have a small user-specified *intrinsic*

[5] We assume each resource is not capacity-limited. The model can be extended to capacity-limited resources as well.

*risk*, denoted $\gamma$, which models the possibility that a resource will not be available indefinitely even if it is not advertised.[6] Let $A_i$ denote the set of channels advertising resource $i$. Then the *total risk* and *total usage* of resource $i$ are, respectively:

$$\Lambda_i = \gamma + \sum_{j \in A_i} \lambda_j \quad \text{and} \quad U_i = \sum_{j \in A_i} u_j. \tag{1}$$

The *yield* of a resource is the product of its usage and lifetime. For example, a proxy with a usage level of 100 user-hours per day lasted 5 days before being blocked, it's yield would be $100 \times 5 = 500$ user-hours. The expected lifetime of a resource is the inverse of its risk, that is, $1/\Lambda_i$. The expected yield is thus $U_i/\Lambda_i$. Our goal is to maximize the expected total system yield, which is simply the sum of the expected yields of each resource. We do this by choosing which resources to advertise on which channels, which requires estimating channels' attracted usage and risk. We assume that it is possible to measure the usage rate attracted by each channel, as described in Section 2. To avoid sharp fluctuations, the estimate may be smoothed using an exponentially-weighted moving average.

### 3.2 Estimating Risk

Because a resource may be advertised using multiple channels, the risk associated with a given channel cannot be sampled directly. When a resource is blocked, we have no way of telling which channel is responsible (in our case, through which channel the censorship authorities found the resource). However from the sample of resource lifetimes, we can compute a maximum likelihood estimate of the risk parameters. Let $t_i$ denote the lifetime of resource $i$. The log-likelihood function of our sample of $m$ resource is:

$$\ell = \log \prod_{i=1}^{m} \Lambda_i e^{-\Lambda_i t_i} = \sum_{i=1}^{m} \big( \log \Lambda_i - \Lambda_i t_i \big). \tag{2}$$

The partial derivatives of $\ell$ with respect to $\lambda_j$ are:

$$\frac{\partial \ell}{\partial \lambda_j} = \sum_{i \in R_j} \big( \Lambda_i^{-1} - t_i \big) \tag{3}$$

The above optimization can be carried out numerically, however a first-order approximation is to attempt to equate $\Lambda_i t_i = 1$. In some cases this may lead to an unsatisfiable system of constraints, so we minimize the sum of squares of $\Lambda_i t_i - 1$, which is analytically tractable. Denote this sum:

$$E = \sum_{i=1}^{m} \big( \Lambda_i t_i - 1 \big)^2, \tag{4}$$

from which we get the system of equations

$$\frac{\partial E}{\partial \lambda_j} = 2 \sum_{i=1}^{m} \big( \Lambda_i t_i - 1 \big) t_i = 0. \tag{5}$$

[6] Alternately, we can view intrinsic risk as a kind of *discount factor* discouraging resource underutilization. Section 3.3 for more on this parameter.

If the number of channels outnumbers the number of resources (that is, $m < n$), the resulting system of equations will be under-constrained. For example, if two channels are used identically (advertising the same set of resources), there is no way to separate their respective risks. To force a unique solution, we introduce another optimization step, minimizing the sum of squares of the risk parameters $\lambda_j$, subject to the linear constraints above. This gives us an estimate of the attracted risk associated with each channel, which we use in deciding which channels to use to advertise resources.

### 3.3 Resource Advertisement Policy

So far we have described how to estimate a channel's usage rate and risk, which we use to choose a resource to advertise on the channel. Our goal is to maximize the expected total yield, which is simply the sum of the expected yields of each resource (recall that the expected yield of a resource is its usage divided by risk, $U_i/\Lambda_i$). When a channel becomes available, we simply choose the resource whose yield would increase the most as a result of being advertised on the new channel. Let $\hat{\jmath}$ denote the index of this channel. The increase in the yield of resource $i$ is then given by

$$\Delta_i = \frac{u_{\hat{\jmath}} + U_i}{\lambda_{\hat{\jmath}} + \Lambda_i} - \frac{U_i}{\Lambda_i} = \frac{u_{\hat{\jmath}}\Lambda_i - U_i\lambda_{\hat{\jmath}}}{(\lambda_{\hat{\jmath}} + \Lambda_i)\Lambda_i}. \tag{6}$$

Our advertisement policy is thus to choose the resource $i$ which maximizes $\Delta_i$.

Note that the numerator of Eq. (6) implies that the increase in yield is positive if and only if $u_{\hat{\jmath}}/\lambda_{\hat{\jmath}} > U_i/\Lambda_i$, in other words, the channel expected yield $u_{\hat{\jmath}}/\lambda_{\hat{\jmath}}$ must be greater than the resource's current expected yield $U_i/\Lambda_i$. This means that, for some low-usage high-risk channels, the best choice is *not to use them at all*. Intuitively, this is because the the risk associated with a channel affects *all* users of a resource, so that the increased risk to the resource is not justified by the additional usage attracted by the channel.

Equation 6 highlights the trade-off inherent in *any* policy of advertising resources, namely the tradeoff between attracting more users or minimizing the risk to the resource. Our "knob" in controlling this trade-off is the intrinsic resource risk parameter $\gamma$. The intrinsic resource risk is the risk of losing a resource *even if we don't advertise it at all*. We conjecture several cases of losing a non-advertised resource such as: a proxy being setup and not well maintained, a proxy server taken offline and re-purposed, and proxy maintainers simply losing interest in our cause.

At the extreme, setting $\gamma = 0$ means that the optimal strategy of advertising $m$ resources is to choose the $m$ highest-yield channels and advertise each resource on exactly one channel. Since in our scenario the resources are much fewer in number than channels, this means that all but the $m$ highest-yield channels will not be used! To be even more concrete, with proxies and registered users, this means that we would assign each registered user their own proxy, choosing the most trustworthy registered users, which is clearly absurd from a practical point of view. The issue is that there is an unstated "discount factor" operating in our scenario: serving 100 users today is better than serving 101 tomorrow. The

effect of the intrinsic resource risk parameter $\gamma$ is akin to such a discount factor: it reduces the expected future yield, stimulating greater resource usage.

## 4 Discussion

*Proximax* is designed to be highly usable, easy to implement, and practical. To achieve these goals the system is left vulnerable to a number of attacks. We discuss some of these attacks and possible counter-measures that would likely impact the usability, implementation complexity, or practicality of *Proximax*, along with a potential optimization that might improve the performance of the system.

**Independence of Adversaries.** To our knowledge, censoring nation states currently act independently, and do not share lists of discovered proxies with each other. Thus some proxies may only be blocked in some countries. Furthermore, certain channels may attract users in the same country due to factors such as common language or personal contacts. *Proximax* could optimize for this situation by detecting specifically which adversaries have blocked a resource and re-assign this resource to another channel that attracts users in a country where the proxy has not been blocked.

**Usage Inflation.** *Proximax* assumes that all usage of a resource is done by legitimate end users that are circumventing a censorship system. However, an adversary can be invited to join our system and inflate their standing (yield) by making dummy connections in order to accrue user-hours. This would cause the system to group the attacker with more resources which they can block. This problem is not specific to *Proximax*, of course; any system which can be fooled into allocating resources to fictitious users is vulnerable. To mitigate this, we can attempt to diversify the user-base by sub-linearly scaling usage credit assigned for attracting very similar users (e.g., users from the same IP address prefix).

**Delayed Blocking.** A smart adversary could infiltrate the system and gather large numbers of proxy addresses and channels and delay acting on this information for weeks or months before blocking them all at once. As part of our system we assume that proxies will be blocked or fail within days or weeks, thus delaying the blocking of proxies to gather more information would likely help prolong the expected longevity of proxies. This in turn would increase the yield of our limited resources, which is the goal of *Proximax*. Thus, we do not consider this much of a risk, since an adversary's user account reputation will drop quickly as she acts upon gathered information.

**Infiltrating Administrator Group.** One of the single points of failure for *Proximax* are the members of the administrative group. If an adversary can infiltrate this group they can gain full information of all proxies that enter the system and our measuring techniques will not be able to detect this properly. If this were to become a problem *Proximax* could be redesigned to limit the amount of information each administrator can access. This could be done by having each administrator setup their own resources and manage a subset of the resource pools. Redesigning the system in this way would make it more complex to aggregate measurement information, but would shield the system from this type of attack.

## 5 Conclusion

In this paper we have presented *Proximax*, an adaptive system for distributing addresses of open proxies that maximizes the yield of user-hours given a limited set of proxy resources. We sketch the different tasks of the system and show how to build an analytical model that uses measurements of the usage rate and blocking risk to intelligently allocate proxy resources among a large number of distribution channels. To our knowledge this it the first system to attempt to build a proxy distribution system that automatically adjusts the resources allocated to each channel and groups registered users together in shared pools of proxies based on similar blocking risk rates. As future work we plan on implementing *Proximax* to gain real measurements that will drive the refinement of our system and analytical model.

## References


1. Wilson, D.: Burmese internet taken down with ddos attack. http://www.techeye.net/internet/burmese-internet-taken-down-with-ddos-attack (2010)
2. Moe, W.: No change as junta clamps down on suu kyi news. http://www.irrawaddy.org/article.php?art_id=20151 (2010)
3. Mackinnon, R.: No quick fixes for internet freedom. http://online.wsj.com/article/SB10001424052748704104104575622080860055498.html (2010)
4. Huffington, A.: Facebook, twitter and the search for peace in the middle east. http://www.huffingtonpost.com/arianna-huffington/facebook-twitter-and-the-_b_788378.html?ir=Technology (2010)
5. : Freegate - dynamic internet technology. http://www.dit-inc.us/freegate
6. : Tor bridges specification. http://gitweb.torproject.org/tor.git?a=blob_plain;hb=HEAD;f=doc/spec/bridges-spec.txt
7. : The proxy fight for iranian democracy. http://www.renesys.com/blog/2009/06/the-proxy-fight-for-iranian-de.shtml (2009)
8. MacKinnon, R.: China's censorship arms race escalates. http://www.circleid.com/posts/20090928_chinas_censorship_arms_race_escalates/ (2009)
9. Lewman, A.: Bridges and china (new thread). http://archives.seul.org/or/talk/May-2010/msg00264.html (2010)
10. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: Proceedings of the 13th USENIX Security Symposium. (August 2004)
11. Dingledine, R., Mathewson, N.: Design of a blocking-resistant anonymity system. https://svn.torproject.org/svn/tor/tags/tor-0_2_0_19_alpha/doc/design-paper/blocking.pdf (2007)
12. : Proxy china - free anonymous surfing. http://www.proxychina.org/
13. : Psiphon design overview 1.0. http://psiphon.ca/documents/Psiphon_Design_Overview_1_0.pdf (2009)
14. Sovran, Y., Libonati, A., Li, J.: Pass it on: Social networks stymie censors. In: 7th International Workshop on Peer-to-Peer Systems (IPTPS). (2008)
15. Mahdian, M.: Fighting censorship with algorithms. In Boldi, P., ed.: FUN. Volume 6099 of Lecture Notes in Computer Science., Springer (2010) 296–306
16. Burnett, S., Feamster, N., Vempala, S.: Chipping away at censorship firewalls with user-generated content. In: Usenix Security,. (August 2010)
17. Riden, J.: How fast-flux service networks work. http://www.honeynet.org/node/132
18. Blakey, G.R.: Racketeer influenced and corrupt organizations act. http://en.wikipedia.org/wiki/Rico_act (2010)