Exploiting Network Structure for Proactive Spam Mitigation

Shobha Venkataraman[*], *Subhabrata Sen*[†], *Oliver Spatscheck*[†], *Patrick Haffner*[†], *Dawn Song*[*]
[*]*Carnegie Mellon University,* [†]*AT&T Research*
shobha@cs.cmu.edu, {*sen,spatsch,haffner*}*@research.att.com, dawnsong@cmu.edu*

Abstract

E-mail has become indispensable in today's networked society. However, the huge and ever-growing volume of spam has become a serious threat to this important communication medium. It not only affects e-mail recipients, but also causes a significant overload to mail servers which handle the e-mail transmission.

We perform an extensive analysis of IP addresses and IP aggregates given by network-aware clusters in order to investigate properties that can distinguish the bulk of the legitimate mail and spam. Our analysis indicates that the bulk of the legitimate mail comes from long-lived IP addresses. We also find that the bulk of the spam comes from network clusters that are relatively long-lived. Our analysis suggests that network-aware clusters may provide a good aggregation scheme for exploiting the history and structure of IP addresses.

We then consider the implications of this analysis for prioritizing legitimate mail. We focus on the situation when mail server is overloaded, and the goal is to maximize the legitimate mail that it accepts. We demonstrate that the history and the structure of the IP addresses can reduce the adverse impact of mail server overload, by increasing the number of legitimate e-mails accepted by a factor of 3.

1 Introduction

E-mail has emerged as an indispensable and ubiquitous means of communication today. Unfortunately, the ever-growing volume of spam diminishes the efficiency of e-mail, and requires both mail server and human resources to handle.

Great effort has focused on reducing the amount of spam that the end-users receive. Most Internet Service Providers (ISPs) operate various types of spam filters [1, 4, 5, 13] to identify and remove spam e-mails before they are received by the end-user. E-mail software on end-hosts adds an additional layer of filtering to remove this unwanted traffic, based on the typical email patterns of the end-user.

Much less attention has been paid to how the large volume of spam impacts the mail infrastructure within an ISP, which has to receive, filter and deliver them appropriately. Spammers have a strong incentive to send large volumes of spam – the more spam they send, the more likely it is that some of it can evade the spam filters deployed by the ISPs. It is easy for the spammer to achieve this – by sending spam using large botnets, spammers can easily generate far more messages than even the largest mail servers can receive. In such conditions, it is critical to understand how the mail server infrastructure can be made to prioritize legitimate mail, processing it preferentially over spam.

In this context, the requirements for differentiating between spam and non-spam are slightly different from regular spam-filtering. The primary requirement for regular spam-filtering is to be conservative in discarding spam, and for this, computational cost is not usually a consideration. However, when the mail server must prioritize the processing of legitimate mail, it has to use a computationally-efficient technique to do so. In addition, in this situation, even an imperfect distinction criterion would be useful, as long as a significant fraction of the legitimate mail gets classified correctly.

In this paper, we explore the potential of using the historical behaviour of IP addresses to predict whether an incoming email is likely to be legitimate or spam. Using IP addresses for classification is computationally substantially more efficient than any content-based techniques. IP address information can also be collected easily and is more difficult for a spammer to obfuscate. Our measurement studies show that IP address information provides a stable discriminator between legitimate mail and spam. We find that good mail servers send mostly legitimate mail and are persistent for significant periods of time. We also find that the bulk of spam comes from IP prefixes that send mostly spam and are also persis-

tent. With these two findings, we can use the properties of *both* legitimate mail and spam together, rather than using the properties of only legitimate mail or only spam, in order to prioritize legitimate mail when needed.

We show that these measurements are valuable in an application where legitimate mail must be prioritized. We focus on the situation when mail servers are overloaded, i.e., they receive far more mail than they can process, even though the legitimate mail received is a tiny fraction of the total received. Since mail typically gets dropped at random when the server is overloaded, and spam can be generated at will, the spammer has an incentive to overload the server. Indeed, the optimal strategy for the spammer is to increase the load on the mail infrastructure to a point where the most spam will be accepted by the server; this kind of behaviour has been observed on the mail servers of large ISPs. In this paper, we show an application of our measurement study to design techniques based on the reputations of IP addresses and their aggregates and demonstrate the benefits to the mail server overload problem.

The contributions of this paper are two-fold. We first perform an extensive measurement study in order to understand some IP-based properties of legitimate mail and spam. We then perform a simulation study to evaluate how we can use these properties to prioritize legitimate mail when the mail server is overloaded.

Our main results are the following:

- We find that a significant fraction of legitimate mail comes from IP addresses that last for a long time, even though a very significant fraction of spam comes from IP addresses that are ephemeral. This suggests that the history of "good" IP addresses, that is, IP addresses that send mostly legitimate mail, could be used for prioritizing mail in spam mitigation.

- We explore *network-aware clusters* as a candidate aggregation scheme to exploit structure in IP addresses. Our results suggest that IP addresses responsible for the bulk of the spam are well-clustered, and that the clusters responsible for the bulk of the spam are persistent. This suggests that network-aware clusters may be good candidates to assign reputations to unknown IP addresses.

- Based on our measurement results, we develop a simple reputation scheme that can prioritize IP addresses when the server is overloaded. Our simulations show that when the server receives many more connection requests than it can process, our policy gives a factor of 3 improvement in the number of legitimate mails accepted.

We note that the server overload problem is just one application that illustrates how IP information could be used for prioritizing email. This information could be used to prioritize e-mail at additional points of the mail server infrastructure as well. However, the kind of structural information that is reflected in the IP addresses may not always be a perfect discriminator between spammers and senders of legitimate mail, and this is, indeed, reflected in the measurements. Such structural IP information could, therefore, be used in combination with other techniques in a general-purpose spam mitigation system, and this information is likely to be useful by itself only when an aggressive and computationally-efficient technique is needed.

The remainder of the paper is structured as follows. We present our analysis of characteristics of IP addresses and network-aware clusters that distinguish between legitimate mail and spam in Sections 2 and 3 respectively. We present and evaluate our solution for protecting mail servers under overload in Section 4. We review related work in Section 5 and conclude in Section 6.

2   Analysis of IP-Address Characteristics

In this section, we explore the extent to which IP-based identification can be used to distinguish spammers from senders of legitimate e-mail based on differences in patterns of behaviour.

2.1   Data

Our data consists of traces from the mail server of a large company serving one of its corporate locations with approximately 700 mailboxes, taken over a period of 166 days from January to June 2006. The location runs a PostFix mail server with extensive logging that records the following: (a) every attempted SMTP connection, with its IP address and time stamp, (b) whether the connection was rejected, along with a reason for rejection, (c) if the connection was accepted, results of additional mail server's local spam-filtering tests, and if accepted for delivery, the results of running SpamAssassin.

Fig. 1(a) shows a daily summary of the data for six months. It shows four quantities for each day: (a) the number of SMTP connection requests made (including those that are denied via blacklists), (b) the number of e-mails received by the mail server, (c) the number of e-mails that were sent to SpamAssassin, and (d) the number of e-mails deemed legitimate by SpamAssassin. The relative sizes of these four quantities on every day illustrate the scale of the problem: spam is 20 times larger than the legitimate mail received. (In our data set, there were 1.4 million legitimate messages and 27 million spam messages in total.) Such a sharp imbalance

indicates the potential of a significant role for applications like maximizing legitimate mail accepted when the server is overloaded: if there is a way to prioritize legitimate mail, the server could handle it much more quickly, because the volume of legitimate mail is tiny in comparison to spam.

In the following analysis, every message that is considered legitimate by SpamAssassin is counted as a legitimate message; every message that is considered spam by SpamAssassin, the mail server's local spam-filtering tests, or through denial by a blacklist is counted as spam.

2.2 Analysis of IP Addresses

We first explore the behaviour of individual IP addresses that send legitimate mail and spam, with the goal of uncovering any significant differences in their behavioral patterns.

Our analysis focuses on the *IP spam-ratio* of an IP address, which we define to be the fraction of mail sent by the IP address that is spam. This is a simple, intuitive metric that captures the spamming behaviour of an IP address: a low spam-ratio indicates that the IP address sends mostly legitimate mail; a high spam-ratio indicates that the IP address sends mostly spam. Our goal is to see whether the historical communication behaviour of IP addresses categorized by their spam-ratios can differentiate between IP addresses of legitimate senders and spammers, for spam mitigation.

As discussed earlier, the differentiation between the legitimate senders and spammers need not be perfect; there are benefits to having even a partial differentiation, especially with a simple, computationally inexpensive feature. For example, in the server overload problem, when all the mail cannot be accepted, a partial separation would still help to increase the amount of legitimate mail that is received.

In the IP-based analysis, we will address the following questions:

- *Distribution by IP Spam Ratio*: What is the distribution of IP addresses by their spam-ratio, and what fraction of legitimate mail and spam is contributed by IP addresses with different spam-ratios?

- *Persistence*: Are IP addresses with low/high spam-ratios present across long time periods? If they are, do such IP addresses contribute to a significant fraction of the legitimate mail/spam?

- *Temporal Spam-Ratio Stability*: Do many of the IP addresses that appear to be good on average fluctuate between having very low and very high spam-ratios?

The answers to these three questions, taken together, give us an idea of the benefit we could derive in using the history of IP address behaviour in spam mitigation. We show in Sec. 2.2.1, that most IP addresses have a spam-ratio of 0% or 100%, and also that a significant amount of the legitimate mail comes from IP addresses whose spam-ratio exceeds zero. In Sec. 2.2.2, we show that a very significant fraction of the legitimate mail comes from IP addresses that persist for a long time, but only a tiny fraction of the spam comes from IP addresses that persist for a long time. In Sec. 2.2.3, we show that most IP addresses have a very high temporal ratio-stability – they do not fluctuate between exhibiting a very low or very high daily spam-ratio over time.

Together, these three observations suggest that *identifying IP addresses with low spam ratios that regularly send legitimate mail* could be useful in spam mitigation and prioritizing legitimate mail. In the rest of this section, we present the analysis that leads to these observations. For concreteness, we focus on how the analysis can help spam mitigation in the server overload problem.
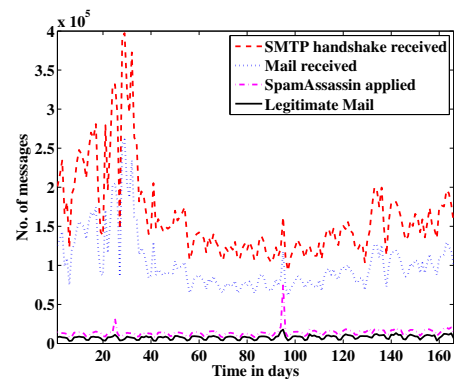
(a) Data characteristics
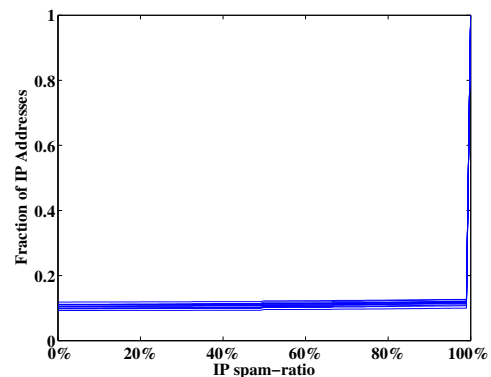
(b) CDFs of IP spam-ratios for many days: each line is a CDF for a different day.

Figure 1: 1(a): Daily summary of the data set over 6 months. 1(b): CDFs of IP spam-ratios for many different days.

2.2.1 Distribution by IP Spam-Ratio

In this section, we explore how the IP addresses and their associated mail volumes are distributed as a function of the IP spam-ratios. We focus here on the spam-ratio computed over a short time period in order to understand the behaviour of IP addresses without being affected by their possible fluctuations in time. Effectively, this analysis shows the limits of the differentiation that could be achieved by using IP spam-ratio, even assuming that IP spam-ratio could be predicted for a given IP address over short periods of time. In this section, we focus on day-long intervals, in order to take into account possible time-of-day variations. We refer to the IP spam-ratio computed over a day-long interval as the *daily spam-ratio*.

Intuitively, we expect that most IP addresses either send mostly legitimate mail, or mostly spam, and that most of the legitimate mail and spam comes from these IP addresses. If this hypothesis holds, then for spam mitigation, it will be sufficient if we can identify the IP addresses as senders of legitimate mail or spammers. To test this hypothesis, we analyze the following two empirical distributions: (a) the distribution of IP addresses as a function of the spam-ratios, and (b) the distribution of legitimate mail/spam as a function of their respective IP addresses' spam-ratio.

We first analyze the distribution of IP addresses by their daily spam-ratios in Fig. 1(b). For each day, it shows the empirical cumulative distribution function (CDF) of the daily spam-ratios of individual IP addresses active on that day. Fig. 1(b) shows this daily CDF for a large number of randomly selected days across the observation period.
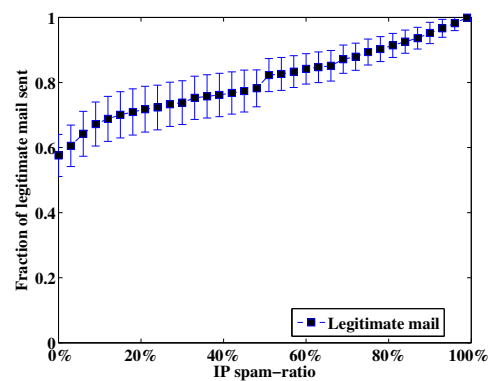
Result 1. Distribution of IP addresses: *Fig. 1(b) indicates: (i) Most IP addresses, send either mostly spam or mostly legitimate mail. (ii) Fewer than $1 - 2\%$ of the active IP addresses have a spam-ratio of between $1\% - 99\%$, i.e., there are very few IP addresses that send a non-trivial fraction of both spam and legitimate mail. (iii) Further, the vast majority (nearly 90%) of IP addresses on any given day generate almost exclusively spam, and have spam-ratios between $99\% - 100\%$.*

The above results indicate that identifying IP addresses with low or high spam-ratios could identify most of the legitimate senders and spammers. In addition, for some applications (e.g., the mail server overload problem), it would be valuable to identify the IP addresses that send the bulk of the spam or the bulk of the legitimate mail, in terms of mail volume. To do so, we next explore how the daily legitimate mail or spam volumes are distributed as a function of the IP spam-ratios, and the resulting implications.

Let I_k denote the set of all IP addresses that have a spam-ratio of at most k. Fig. 2 examines how the volume

of legitimate mail and spam sent by the set I_k depends on the spam-ratio k. Specifically, let $L_i(k)$ and $S_i(k)$ be the fractions of the total daily legitimate mail and spam that comes from all IPs in the set I_k, on day i. Fig. 2(a) plots $L_i(k)$ averaged over all the days, along with confidence intervals. Fig. 2(b) shows the corresponding distribution for the spam volume $S_i(k)$.

Result 2. Distribution of legitimate mail volume: *Fig. 2(a) shows that the bulk of the legitimate mail (nearly 70% on average) comes from IP addresses with a very low spam-ratio ($k \leq 5\%$). However, a modest fraction (over 7% on average) also comes from IP addresses with a high spam-ratio ($k \geq 80\%$).*

(a) Legitimate mail
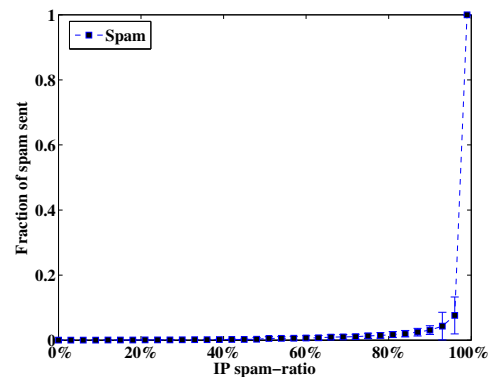
(b) Spam

Figure 2: Legitimate mail and spam contributions as a function of IP spam-ratio.

Result 3. Distribution of spam volume: *Fig. 2(b) indicates that almost all (over 99% on average) of the spam sent every day comes from IP addresses with an extremely high spam-ratio (when $k \geq 95\%$). Indeed, the contribution of the IP addresses with lower spam-ratios ($k \leq 80\%$) is a tiny fraction of the total.*

We observe that the distribution of legitimate mail volume as a function of the spam-ratio k is more diffused

than the distribution of spam volume. There are two possible explanations for such behaviour of the legitimate senders. First, spam-filtering software tends to be conservative, allowing some spam to marked as legitimate mail. Second, a lot of legitimate mail tends to come from large mail servers that cannot do perfect outgoing spam-filtering. These mail servers may, therefore, have a slightly higher IP spam-ratio, and this would cause the distribution of legitimate mail to be more diffused across the spam-ratio.

Together, the above results suggest that the IP spam-ratio may be a useful discriminating feature for spam mitigation As an example, assume that we have a classification function that accepted (or prioritized) all IP addresses with a spam-ratio of at most k and rejected all IP addresses with a higher spam-ratio. Then, if we set $k = 95\%$, we could accept (or prioritize) nearly all the legitimate mail, and no more than 1% of the spam. However, such a classification function requires perfect knowledge of every IP address's daily spam-ratio every single day, and in reality, this knowledge may not be available.

Instead, our approach is to identify properties that occur over longer periods of time, and are useful for predicting the current behaviour of an IP address based on long-term history, and these properties are incorporated into classification functions. The effectiveness of such history-based classification functions for spam mitigation depends on the extent to which IP addresses long-lived, how much of the legitimate email or spam are contributed by the long-lived IP addresses, and to what extent the spam-ratio of an IP address varies over time. Sec. 2.2.2 and Sec. 2.2.3 explore these questions.

For the following analysis, we focus on the spam-ratio of each individual IP address, computed over the entire data set, since we are interested in its behaviour over its lifetime. We refer to this as the *lifetime spam-ratio* of the IP address. We show the presence of two properties in this analysis: (i) a significant fraction of legitimate mail comes from good IP addresses that last for a long time (*persistence*), and (ii) IP addresses that are good on average tend to have a low spam-ratio each time they appear (*temporal stability*). These two properties directly influence how effective it would be to use historical information for determining the likelihood of spam coming from an individual IP address.
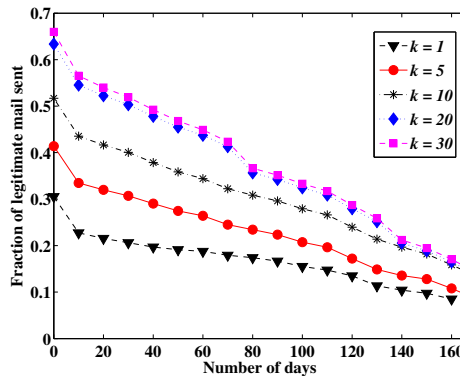
2.2.2 Persistence

Due to the community structure inherent in non-spam communication patterns, it seems reasonable that most of the legitimate mail will originate from IP addresses that appear and re-appear. Previous studies have also indicated that most of the spam comes from IP addresses that

are extremely short-lived. These suggest the existence of a potentially significant difference in the behaviour of senders of legitimate mail and spammers with respect to persistence. We next quantify the extent to which these hypotheses hold, by examining the persistence of individual IP addresses.

Our methodology for understanding the persistence behavior of IP addresses is as follows: we consider the set of all IP addresses with a low lifetime spam-ratio and examine how much legitimate mail they send, as well as how much of the legitimate mail is sent by IP addresses that are present for a long time. Such an understanding can indicate the potential of using a whitelist-based approach for prioritizing legitimate mail. If, for instance, the bulk of the legitimate mail comes from IP addresses that last for a long time, we could use this property to prioritize legitimate mail from long-lasting IP addresses with low spam-ratios.
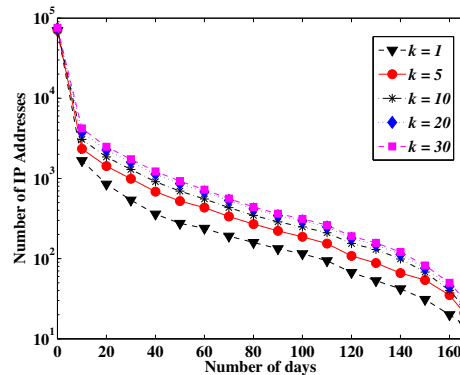
(a) Number of k-good IP addresses present for x or more days

(b) Fraction of legitimate mail sent by k-good IP addresses present for x or more days

Figure 3: Persistence of k-good IP addresses.

For this analysis, we use the following two definitions.

Definition 1. *A k-**good IP address** is an IP address whose lifetime spam-ratio is at most k. A k-**good set** is the set of all k-good IP addresses. Thus, a 20-good set*

is the set of all IP addresses whose lifetime spam-ratio is no more than 20%.

We compute (a) the number of k-good IP addresses present for at least x distinct days, and (b) the fraction of legitimate mail contributed by k-good IP addresses that are present in at least x distinct days. [1] Fig. 3(a) shows the number of IP addresses that appear in at least x distinct days, for several different values of k.

Fig. 3(b) shows the fraction of the total legitimate mail that originates from IP addresses that are in the k-good set and appear in at least x days, for each threshold k.

Most of the IP addresses in a k-good set are not present very long, and the number of IP addresses falls quickly, especially in the first few days. However, their contribution to the legitimate mail drops much more slowly as x increases. The result is that the few longer-lived IPs contribute to most of the legitimate mail from a k-good set. For example, only 5% of all IP addresses in the 20-good set appear at least 10 distinct days, but they contribute to almost 87% of all legitimate mail for the 20-good set. If the k-good set contributes to a significant fraction of the legitimate mail, then the few longer-lived IP addresses also contribute significantly to the total legitimate mail. For instance, IP addresses in the 20-good set contribute to 63.5% of the total legitimate mail received. Only 2.1% of those IP addresses are present for at least 30 days, but they contribute to over 50% of the total legitimate mail received.

Result 4. Distribution of legitimate mail from persistent k-good IPs: *Fig. 3 indicates that (i) IP addresses with low lifetime spam ratios (small k) tend to contribute a major proportion of the total legitimate email, and (ii) only a small fraction of the IP addresses with a low lifetime spam-ratio addresses appear over many days, but they contribute to a significant fraction of the legitimate mail.*

The graphs also reveal another trend: the longer an IP address lasts, the more stable is its contribution to the legitimate mail. For example, 0.09% of the IP addresses in the 20-good set are present for at least 60 days, but they contribute to over 40% of the total legitimate mail received. From this, we can infer that there were an additional 1.2% of IP addresses in the 20-good set that were present for 30-59 days, but they only contributed to 10% of the total legitimate mail received.

---

[1] Our analysis considers persistence of IP addresses only in our data set, i.e., it considers whether the IP address has sent mail for x days to our mail server. These IP addresses may have sent mail to other mail servers on more days, and combining data across multiple different mail servers may give a better picture of stability of IP addresses sending mail. Nevertheless, in this work, we focus on the persistence in one data set, as it highlights behavioural differences due to community structure present within a single vantage point.

Fig. 4 presents a similar analysis of persistence for IP addresses with a high lifetime spam-ratio. Like the k-good IP addresses and k-good sets, we define k-bad IP addresses and k-bad sets.

Definition 2. *A k-bad IP address is an IP address that has a lifetime spam-ratio of at least k. A k-bad set is the set of all k-bad IP addresses.*
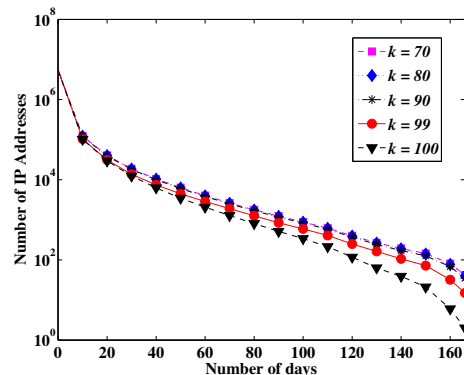
(a) No. of k-bad IP addresses present in x or more days
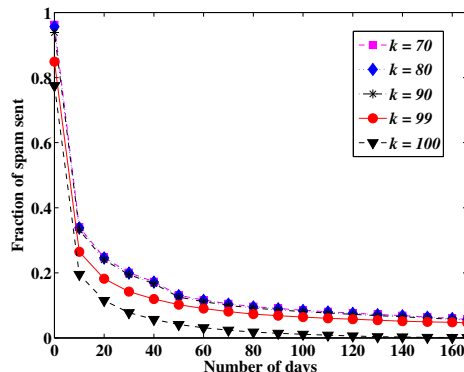
(b) Fraction of spam sent by k-bad IP addresses present in x or more days

Figure 4: Persistence of k-bad IP addresses.

Fig. 4(a) presents the number of IP addresses in the k-bad set that are present in at least x days, and Fig. 4(b) presents the fraction of the total spam sent by IP addresses in the k-bad set that are present in least x days.

Result 5. Distribution of spam from persistent k-bad IPs: *Fig. 4 indicates that (i) IP addresses with high lifetime spam ratios (large k) tend to contribute almost all of the spam, (ii) most of these high spam-ratio IPs are only present for a short time (this is consistent with the finding in [19]) and account for a large proportion of the overall spam, and (iii) the small fraction of these IPs that do last several days contribute a non-trivial fraction of the overall spam; however, a much larger fraction of spam comes from IP addresses that are not present for*

very long. As in the case of the k-good IP addresses, the spam contribution from the k-bad IP addresses tends to get more stable with time.

So, for instance, we can see from Fig. 4 that only 1.5% of the IP addresses in the 80-bad set appear in at least 10 distinct days, and these contribute to 35.4% of the volume of spam from the 80-bad set, and 34% of the total spam. The difference is more pronounced for 100-bad IP addresses: 2% of the 100-bad IP addresses appear for 10 or more distinct days, and contribute to 25% of the total spam volume.

The results of this section have implications in designing spam filters, especially for applications where the goal is to prioritize legitimate mail rather than discard spam. While spamming IP addresses that are present sufficiently long can be blacklisted, the scope of a purely blacklisting approach is limited. On the other hand, a very significant fraction of the legitimate mail can be prioritized by using the history of the senders of legitimate mail.

2.2.3 Temporal Stability

Next, we seek to understand whether IP addresses in the k-good set change their daily spam-ratio dramatically over the course of their lifetime. The question we want to answer is: of the IP addresses that appear in a k-good set (for small values of k), what fraction of them have ever had "high" daily spam-ratios, and how often do they have "high" spam-ratios? Thus, we want to understand the *temporal stability* of the spam-ratio of IP addresses in k-good sets. In this section, we focus on k-good IP addresses; the results for the k-bad IP addresses are similar.
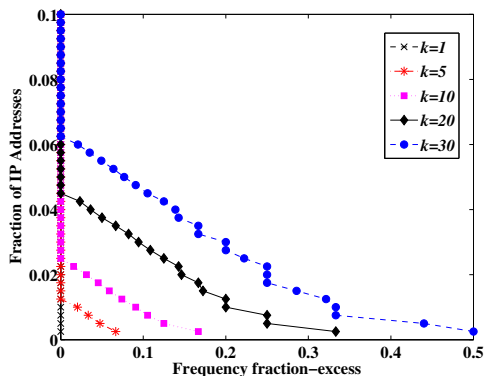

Figure 5: Temporal stability of IP addresses in k-good sets, shown by CCDF of frequency-fraction excess.

We compute the following metric: for each IP address in a k-good set, we count how often its daily spam-ratio exceeds k (and normalize this count by the num-

ber of days it appears). We define this quantity to be the *frequency-fraction excess* of the IP address, for the k-good set. We plot the complementary cdf (CCDF) of the *frequency-fraction excess* of all IP addresses in the k-good set. [2] Intuitively, the distribution of the frequency-fraction excess is a measure of how many IP addresses in the k-good set exceed k, and how often they do so.

Fig. 5 shows the CCDF of the frequency-fraction excess for several k-good sets. It shows that the majority of the IP addresses in each k-good set have a frequency-fraction excess of 0, and that 95% of the k-good IP addresses have a frequency-fraction excess of at most 0.1.

We explain the implications of Fig. 5 to the temporal stability of the spam-ratio of IP addresses with an example. We focus on the k-good set for $k = 20$: this is the set of IP addresses whose lifetime spam-ratio is bounded by 20%. We note that the frequency-fraction excess is 0 for 95% of the 20-good IP addresses. This implies that 95% of IP addresses in this k-good set do not send more than 20% spam *any* day, i.e., every time they appear, they have a daily spam-ratio of at most 20%. We also note that fewer than 1% of the IP addresses in this k-good set have a frequency-fraction excess larger than 0.2.

Thus, for many k-good sets with small k-values, only a few IP addresses have a significant frequency-fraction excess, i.e., very few IP addresses in those sets exceed the value k often. Since they would need to exceed k often to change their spamming behaviour significantly, it follows that most IP addresses in the k-good set do not change their spamming behaviour significantly.

In addition, the frequency-fraction excess is perhaps too strict a measure, since it is affected even if k is exceeded slightly. We also compute a similar measure that increases only when k is exceeded by 5%. No more than 0.01% of IP address in the k-good set exceed k by 5%, for any $k \leq 30\%$. Since we are especially interested in the temporal stability of IP addresses that appear often, we compute also the frequency-fraction excess distribution for IP addresses that appear for 10, 20, 40 and 60 days. In each case, almost no IP address exceeds k by more than 5%, for any $k \leq 30\%$.

We summarize this discussion in the following result.

Result 6. Temporal stability of k-good IPs: *Fig. 5 shows that most IP addresses in k-good sets (for low k, e.g., $k \leq 30\%$) do not exceed k often; i.e., most k-good IP addresses have low spam-ratios (at most k) nearly every day.*

With the above result, we can analyze the behaviour of k-good sets of IP addresses, constructed over their entire lifetime, and their behaviour in shorter time intervals.

---

[2]That is, we plot the fraction of IP addresses in the k-good set whose frequency-fraction excess is at least x. The y-axis of the plot is restricted for readability.

The analysis of these three properties of IP addresses indicates that a significant fraction of the legitimate mail comes from IP addresses that persistently appear in the traffic. These IP addresses tend to exhibit stable behaviour: they do not fluctuate significantly between sending spam and legitimate mail. These results lend weight to our hypothesis that spam mitigation efforts can benefit by preferentially allocating resources to the stable and persistent senders of legitimate mail. However, there is still a substantial portion of the mail that cannot be accounted for through only IP address-based analysis. In the next section, we focus on how to account for this mail.

3 Analysis of Cluster Characteristics

So far, we have analyzed whether the historical behaviour of individual IP addresses can be used to distinguish between senders of legitimate mail and spammers. However, if we only consider the history of individual IP addresses, we cannot determine whether a new, previously unseen, IP address is likely to be a spammer or a sender of legitimate mail. If there are many such IP addresses, then, in order to be useful, any prioritization scheme would need to assign these new IP addresses appropriate reputations as well. Indeed, in Sec. 2.2.2, we found that most IP addresses sending mail are short-lived and that such short-lived IPs account for a significant proportion of both legitimate mail and spam. Any prioritization scheme would thus need to be able to find reputations for these IP addresses as well.

To address this issue, we explore whether coarser aggregations of IP addresses exhibit more persistence and afford more effective discriminatory power for spam mitigation. If such aggregations of IP addresses can be found, the reputation of an unseen IP address could be *derived* from the historical reputation of the aggregation they belong to.

We focus on IP aggregations given by *network-aware clusters* of IP addresses [15]. Network-aware clusters are sets of unique network IP prefixes collected from a wide set of BGP routing table snapshots. In this paper, an IP address belongs to a network-aware cluster if the longest prefix match of the IP address matches the prefix associated with the cluster. In the reputation mechanisms we explore in Sec. 4, an IP address derives the reputation of the network-aware cluster that it belongs to. We use network-aware clustering because these clusters represent IP addresses that are close in terms of network topology and do, with high probability, represent regions of the IP space that are under the same administrative control and share similar security and spam policies [15].

In this section, we present measurements suggesting that network-aware clusters of IP addresses may provide

a good basis for reputation-based classification of IP addresses. We focus on the following questions:

- *Granularity*: Does the mail originating from network-aware clusters consist of mostly spam or mostly legitimate mail, so that these clusters could be useful as a reputation-granting mechanism for IP addresses?

- *Persistence*: Do individual network-aware clusters appear (i.e., do IP addresses belonging to the clusters appear) over long periods of time, so that network-aware clusters could potentially afford us a useful mechanism to distinguish between different kinds of ephemeral IP addresses?

As in the IP-address case, we adopt the spam-ratio of a network-aware cluster as the discriminating feature of clusters and examine whether clusters with low/high spam-ratios are granular and persistent.

Before examining these two properties in detail, we first summarize our analysis of the properties with respect to which clusters behave as IP addresses do: *clusters turn out to be at least as (and usually more) temporally stable as IP addresses* (similar to the IP address behaviour explored in Sec. 2.2.3), which is the expected behaviour; *the distribution of clusters by daily cluster spam-ratio is similar to the distribution of IP addresses by IP spam-ratio* (similar to the IP address behaviour explored in Sec. 2.2.1).

3.1 Cluster Granularity

For network-aware clustering of IP addresses to be useful, the clusters need to be sufficiently homogeneous in terms of their legitimate mail/spam behavior so that the cluster information can be used to separate the bulk of legitimate mail from the bulk of spam. Recall that with the IP addresses, we analyzed the extent to which IP spam-ratios could be used to identify the IP addresses sending the bulk of legitimate mail and spam. Here, we analyze whether, instead of an IP's individual spam-ratio, the spam-ratio of the parent cluster can be used for the same purpose.

To do so, we need to understand how well the cluster spam-ratio approximates the IP spam-ratio. In our context, we focus on the following question: can we still distinguish between the IP addresses that send the bulk of the legitimate mail and the bulk of the spam? If we can, within a margin of error, it would suggest that cluster-level analysis is nearly as good as IP-level analysis.

For the analysis here, we determine the spam-ratio of each cluster by analyzing the mail sent by all IP addresses belonging to that cluster and assign to IP addresses the spam-ratios of their respective clusters. In
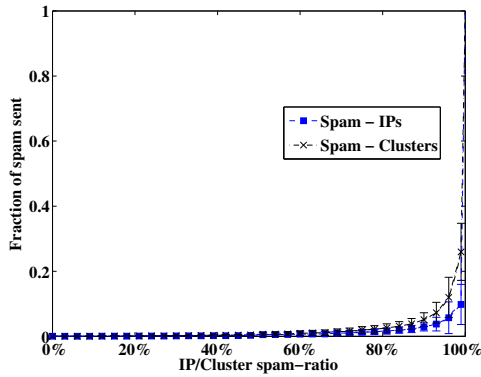
(a) Fraction of spam sent by clusters & IPs, as a function of cluster & IP spam-ratios.
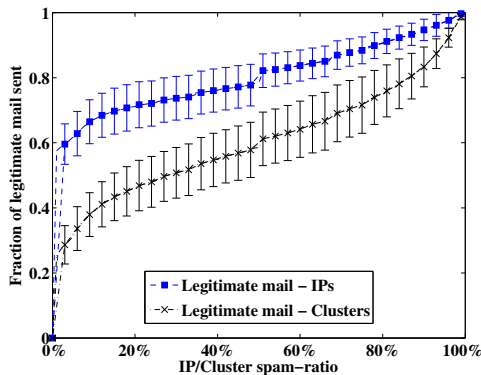
(b) Legitimate mail sent by clusters & IPs, as a function of cluster & IP spam-ratios.

Figure 6: Penalty of using cluster-level analysis.

the rest of this discussion, we will refer to legitimate mail/spam sent by IP addresses belonging to a cluster as the legitimate mail/spam *sent by* or *coming from* that cluster. As with the IP-based analysis, we examine how the volume of legitimate mail and spam from IP addresses is distributed as a function of their cluster spam-ratios. To understand the additional error imposed by using the cluster spam-ratio, we compare it with how those volumes are distributed as a function of the IP spam-ratio.

Fig. 6(a) shows how the spam sent by IP addresses with a cluster or IP spam-ratio of at most k varies with k. Specifically, on day i, let $CS_i(k)$ and $IS_i(k)$ be the fraction of spam sent by the IP addresses with a cluster spam-ratio (and IP spam-ratio, respectively) of at most k. Fig. 6(a) plots $CS_i(k)$ and $IS_i(k)$ averaged over all the days in the data set, as a function of k, along with confidence intervals.

Result 7. Distribution of spam with cluster and IP spam-ratios: *Fig. 6(a) shows that almost all (over* 95%*) of the spam every day comes from IPs in clusters with a very high cluster spam-ratio (over* 90%*). A similar frac-*

tion (over 99% *on average) of the spam every day comes from IP addresses with a very high IP spam-ratio (over* 90%*).*

This suggests that spammers responsible for a high volume of the total spam may be closely correlated with the clusters that have a very high spam-ratio. The graph indicates that if we use a spam-ratio threshold of $k \leq 90\%$ for spam mitigation, then using the IP spam-ratio rather than the corresponding cluster spam-ratio as the discriminating feature would increase the amount of spam identified by less than 2%. This suggests that cluster spam-ratios are a good approximation to IP spam-ratios for identifying the bulk of the spam sent.

We next consider how legitimate mail is distributed with the cluster spam-ratios and compare it with IP spam-ratios (Fig. 6(b)). We compute the following metric: Let $CL_i(k)$ and $IL_i(k)$ be the fraction of legitimate mail sent by IPs with cluster and IP spam-ratios of at most k on day i. Fig. 6(b) plots $CL_i(k)$ and $IL_i(k)$ averaged over all the days in the data set as a function of k, along with confidence intervals.

Result 8. Distribution of legitimate mail with cluster and IP spam-ratios: *Fig. 6(b) shows that a significant amount of legitimate mail is contributed by clusters with both low and high spam-ratios. A significant fraction of the legitimate mail (around* 45% *on average) comes from IP addresses with a low cluster spam-ratio (* $k \leq 20\%$ *). However, a much larger fraction of the legitimate mail (around* 70%, *on average) originates from IP addresses with a similarly low IP spam-ratio.*

The picture here, therefore, is much less promising: even when we consider spam-ratios as high as $30-40\%$, the cluster spam-ratios can only distinguish, on average, around 50% of the legitimate mail. By contrast, IP spam-ratios can distinguish as much as 70%. This suggests that IP addresses responsible for the bulk of legitimate mail are much less correlated with clusters of low spam-ratio.

We can then make the following conclusion: suppose we use a classification function to accept or reject IP addresses based on their cluster spam-ratio. What additional penalty would we incur over a similar classification function that used the IP address's own spam-ratio? Fig. 6(b) suggests that, if the threshold is set to 90% or higher, we incur very little penalty in both legitimate mail acceptance and spam. However, if the threshold is set to $30-40\%$, we may incur as much as a 20% penalty in doing so.

However, there are two additional ways in which such a classification function could be enhanced. First, as we have seen, the bulk of the legitimate mail does come from persistent k-good IP addresses. This suggests that we could potentially identify more legitimate mail by considering the persistent k-good IP addresses *in addition*

to cluster-level information. Second, for some applications, the correlation between high cluster spam-ratios and the bulk of the spam may be sufficient to justify using cluster-level analysis. For example, under the existing distribution of spam and legitimate mail, even a high cluster spam-ratio threshold would be sufficient to reduce the total volume of the mail accepted by the mail server. This is exactly the situation in the server overload problem and we see the effect in the simulations in Sec. 4.
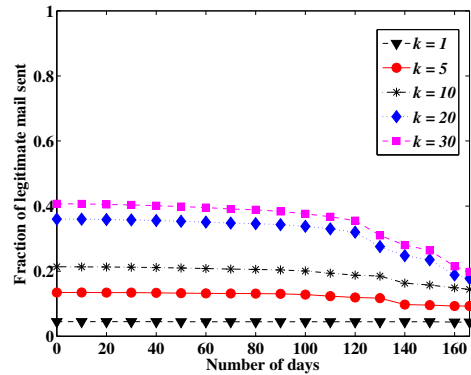
3.2 Persistence

Next, we explore how persistent the network-aware clusters are, just as we did for the IP addresses. We define a cluster to be *present* on a day if at least one IP address that belongs to that cluster appears that day. We reported earlier that we found the clusters themselves to be at least as (and usually more) temporally stable as IP addresses. Our next goal is to examine how much of the total legitimate mail/spam the long-lived clusters contribute.

As in Sec. 2.2.2, we will define k-good and k-bad clusters; to do that, we use the *lifetime cluster spam-ratio*: the ratio of the total spam sent by the cluster to the total mail sent by it over its lifetime.

Definition 3. *A k-good cluster is a cluster of IP addresses whose lifetime cluster spam-ratio is at most k. The k-good cluster-set is the set of all k-good clusters. A k-bad cluster is a cluster of IP addresses whose lifetime cluster spam-ratio is at least k. The k-bad cluster-set is the set of all k-bad clusters.*

Fig. 7(a) examines the legitimate mail sent by k-good clusters for small values of k. We first note that the k-good clusters (even when k is as large as 30%) contribute less than 40% of the total legitimate mail; this is in contrast to, for instance, 20-good IP addresses that contributed to 63.5% of the total legitimate mail. However, we note the contribution from long-lived clusters is far more than from long-lived individual IPs. The difference from Fig. 3(b) is striking: e.g., k-good clusters present for 60 or more days contribute to nearly 99% of the legitimate mail from the k-good cluster set. So, any cluster accounting for a non-trivial volume of legitimate mail is present for at least 60 days. Indeed, the legitimate mail sent by k-good clusters drops to 90% of k-good cluster-set's total only when restricted to clusters present for 120 or more days; by contrast, for individual IP addresses, the legitimate mail contribution dropped to 87% of the 20-good set's total after just 10 days.

Fig. 7(b) presents the same analysis for k-bad clusters. Again, there are noticeable differences from the k-bad IP addresses, and also from the k-good clusters. A much larger fraction of spam comes from long-lived clusters than from long-lived IPs in Fig. 4(b). For example, over

(a) Fraction of legitimate mail sent by k-good clusters that appear in at least x days
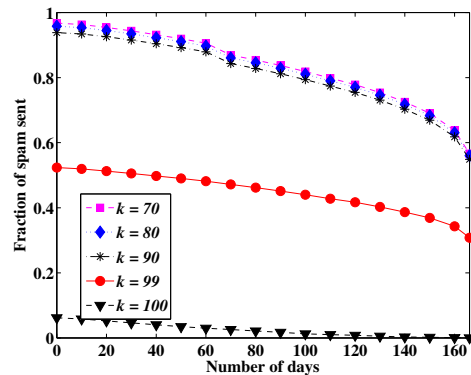
(b) Fraction of spam sent by k-bad clusters that appear in at least x days

Figure 7: Persistence of network-aware clusters.

92% of the total spam is contributed by 90-bad clusters present for at least 20 days. This is in sharp contrast with the k-bad IP addresses, where only 20% of the total spam comes from IP addresses that last 20 or more days. We also note that the 90-bad cluster-set contributes to nearly 95% of the total spam. Thus, in contrast to the legitimate mail sent by k-good cluster-sets, the bulk of the spam comes from the k-bad cluster-sets with high k.

Result 9. Distribution of mail from persistent clusters: *Fig. 7 shows that the clusters that are present for long periods with high cluster spam-ratios contribute the overwhelming fraction of the spam sent, while those present for long periods with low cluster spam-ratios contribute a smaller, though still significant, fraction of the legitimate mail sent.*

The above result suggests that network-aware clustering can be used to address the problem of transience of IP addresses in developing history-based reputations of IP addresses: even if individual IP addresses are ephemeral, their (possibly collective) history would be useful in assigning reputations to other IP addresses originating

from the same cluster.

4 Spam Mitigation under Mail Server Overload

In the previous section, we have demonstrated that there are significant differences in the historical behaviour of IP addresses that send a lot of spam, and those that send very little. In this section, we consider how these differences in behaviour could be exploited for spam mitigation.

Our measurements have shown that senders of legitimate mail demonstrate significant stability and persistence, while spammers do not. However, the bulk of the high volume spammers appear to be clustered well within many persistent network-aware clusters. Together, these suggest that we can design techniques based on the historical reputation of an IP address and the cluster to which it belongs. However, because mail rejection mechanisms necessarily need to be conservative, we believe that such a reputation-based mechanism is primarily useful for prioritizing legitimate mail, rather than actively discarding all suspected spammers.

As an application of these measurements, we now consider the mail-server overload problem described in the introduction. In this section, we demonstrate how the problem could be tackled with a reputation-based mechanism that exploits these differences in behaviour. In Sec. 4.1, we explain the mail-server overload problem in more detail. In Sec. 4.2, we explain our approach, describing the mail server simulation and algorithms that we use, and in Sec. 4.3, we present an evaluation showing the performance improvement gained using these differences in behaviour.

We emphasize that this simulation study is intended to demonstrate the potential of using these behavioural differences in the legitimate mail and spam for prioritizing exclusively by IP addresses. However, it is *not* intended to be comparable to content-based spam filtering. We also note that these differences in behaviour could be applied in other ways as well and at other points in the mail processing as well. The quantitative benefits that we achieve may be specific to our application and may be different in other applications.

4.1 Server Overload Problem

The problem we consider is the following: When the mail server receives more SMTP connections than it can process in a time interval, how can it selectively accept connections to maximize the acceptance of legitimate mail? That is, the mail server receives a sequence of connection requests from IP addresses every second, and each connection will send mail that is either legitimate or spam. Whether the IP address sends spam or legitimate mail in that connection is not known at the time of the request, but is known after mail is processed by the spam filter. The mail server has a finite capacity of the number of mails that can be processed in each time interval, and may choose the connections it accepts or rejects. The goal of the mail server is to selectively accept connections in order to maximize the legitimate mail accepted.

We note that spammers have strong incentive to cause mail servers to overload, and illustrate this with an example. Assume that a mail server can process 100 emails per second, that it will start dropping new incoming SMTP connections when its load reaches 100 emails per second, and that it crashes if the offered load reaches 200 emails per second. Assume also that 20 legitimate emails are received per second. A spammer could increase the load of the mail server to 100% by sending 80 emails per second which would be all received by the mail server. Alternatively, the spammer could also increase the load to 199%, by sending 179 spam emails per second, and now nearly half the requests would not be served. If the mail server is unable to distinguish between the spam requests and the legitimate mail requests, it drops connections at random, and the spammer will be able to successfully get through 89 spam emails per second to the mail server, as compared to the 80 in the previous case.

Thus, the optimal operation point of a spammer, assuming that he has a large potential sending capacity, is not the maximum capacity of the mail server but the maximum load before the mail server will crash. This observation indicates that the approach of throwing more resources at the problem would only work if the mail server capacity is increased to exceed the largest botnet available to the spammer. This is typically not economically feasible and a different approach is needed.

The results in Sec. 2 and Sec. 3 suggest that there may be a history-based reputation function R, that relates IP addresses to their likelihood of sending spam. Thus, for example, if $R(i)$ is the probability that an IP address i sends legitimate mail, then maximizing the quantity $\sum R(i)$ would maximize the expected number of accepted legitimate mail. If the reputation function R were known, this problem would be similar to admission control and deadline scheduling; however, in our case, R is not known.

In this work, we choose one *simple* history-based reputation function and demonstrate that it performs well. We reiterate that our goal is *not* to explore the space of the reputation functions or to find the best reputation function. Rather, our goal is to demonstrate that they could potentially be used to increase the legitimate mail accepted when the mail-server is overloaded. In addition, our goal is to preferentially accept e-mails from certain IP addresses *only* when the mail servers are overloaded

– we would like to minimize the impact on mail servers when they are not overloaded. A poor choice of R will then not impact the mail server under normal operation.

The techniques and the reputation functions that we choose address concerns that are different from those addressed by standard IP-based classification techniques like blacklisting and greylisting, as neither blacklisting nor greylisting would directly solve the server overload problem. Blacklisting has well-known issues: building a blacklist takes time and effort, most IP addresses that send spam are observed to be ephemeral, appearing very few times, and many of them are not even present in any single blacklist.

While greylisting is an attractive short-term solution that has been observed to work quite well in practice, it is not robust to spammer evasion, since spammers could simply mimic the behaviour of a normal mail server. Greylisting aims to optimize a different goal – its goal is to delay the mail in the hope that a spam signature is generated in the mean time, so that spam can be distinguished from non-spam; however, delaying the mail does not reduce the overall server load, since the spammer can always return to send more mail, and computing a content-based spam signature would continue to be as expensive. Indeed, greylisting gives spammers even more incentive to overload mail servers by re-trying after a specified time period.

Our techniques for the server overload problem provide an additional layer of information when compared to blacklisting and greylisting. It may be possible to use the IP structure information to enhance greylisting, to decide, at finer granularities and with soft thresholding, which IP addresses to deny.

4.2 Design and Algorithms

Today, when mail servers experience overload, they drop connections greedily: the server accepts all connections until it is at maximum load, and then refuses all connection requests until its load drops below the maximum. We aim to improve the performance under overload by using information in the structure of IP addresses, as suggested by the results in Sec. 2 and Sec. 3. At a high-level, our approach is to obtain a history of IP addresses and IP clusters, and use it to select the IP addresses that we prioritize under overload. To explore the potential benefits of this approach, we simulate the mail server operation and allow some additional functionality to handle overload.

To motivate our simulation, we describe briefly the way many mail servers in corporations and ISPs operate. First, the sender's mail server or a mail relay tries to connect to the receiving mail server via TCP. The receiving mail server accepts the connection if capacity is available, and then the mail servers perform the SMTP handshake and transfer the email. The receiving mail server stores the email to disk and adds it to the spam processing queue. For each e-mail on the queue, the receiving mail server then performs content-based spam filtering [3, 1] which is typically the most expensive part of email processing. After this, the spam emails are dropped or delivered to a spam mailbox, and the good emails are delivered to the inbox of the recipient.

In our simulation we simplify the mail server model, while ensuring that it is still sufficiently rich to capture the problem that we explore. We believe that our model is sufficiently representative for a majority of mail server implementations used today; however, we acknowledge that there are mail server architectures in use which are not fully captured in our model. In the next section, we describe the simulation model in more detail.

4.2.1 Mail Server Simulation

We simulate mail-server operation in the following manner:

- *Phase 1:* When the mail server receives an SMTP connection request, it may decide whether or not to accept the connection. If it decides to accept the connection, the incoming mail takes t time units to be transferred to the mail server. Thus, if a server can accept k connection requests simultaneously, it behaves like a k-parallel processor in this phase. We do so because this phase models the SMTP handshake and transfer of mail, and therefore, it needs to model state for each connection separately.

- *Phase 2:* Once the mail has been received, it is added to a queue for spam filtering and delivery to the receiving mailbox if any. At each time-step, the mail server selects mails from this queue and processes them; the number of mails chosen depend on the mail server's capacity and the cost of each individual mail. Here, since we model computation cycles, a sequential processing model suffices. The mail server has a timeout: it discards any mail that has been in the queue for more than m time units. If the load has sufficient fluctuation, a large timeout would be useful, but we want to minimize timeout since email has the expectation of being timely.

We assume that the cost of denying/dropping a request is 0, the cost of processing the SMTP connection is α fraction of its total cost, and the cost of the remainder is $1 - \alpha$ fraction of the total cost. We also allow Phase 1 of the mail server simulator to have α fraction of the server's computational resources, and Phase 2 to have the remainder. Since the content-based analysis is typ-

ically the most expensive part of processing a message, we expect that α is likely to be small.

This two-phase simulation model allows for more flexibility in our policy design, since it opens the possibility of dropping emails which have already been received and are awaiting spam filtering without wasting too many resources.

4.2.2 Policies

Next, we present the prioritization/drop policies that we implemented and evaluated on the mail server simulator. In this simulation model, the default mail-server action corresponds to the following: at each time-interval, the server accepts incoming requests in the order of arrival, as long as it is not overloaded. Once mail has been received, the server processes the first mail in the queue, and discards any mail that has exceeded its timeout. We refer to this as the *greedy* policy.[3]

The space of policy options that a mail-server is allowed to operate determine the kinds of benefits it can get. In this problem, one natural option for the mail server is to decide immediately whether to accept or reject a connection request. However, such a policy may be quite sensitive to fluctuation in the workload received at the mail server. Another option may be to reject some e-mails *after* the SMTP connection has been accepted, but *before* any spam-filtering checks or content-based analysis (such as spam-filtering software) has been applied. Note that content-based analysis typically is the most computationally expensive part of receiving mail. Thus, with this option, the mail server may do a small amount of work for some additional emails that eventually get rejected, but is less affected by the fluctuation of mail arrival workload. We restrict the space of policy options to the time before *any* content-based analysis of the incoming mail is done.

To solve the mail-server overload problem, we implement the following policies at the two phases:

- *Phase-1 policy*: The policy in Phase 1 is designed to preferentially accept IP addresses with a good reputation when the server is near maximum load: as the server gets closer to overload, the policy only accepts IP addresses with better and better reputations. The policy itself is more complex, since it needs to consider the expected legitimate mail workload, and yet not stay idle too long. We therefore leave exact details to the appendix. In addition, when the load is below some percentage (we choose 75%) of the

total capacity, the server accepts all mail: this way, it minimizes impact on normal operation of the mail server. [4]

- *Phase-2 policy*: The scheduling policy here is easier to design, since the queue has some knowledge of what needs to be processed. Even a simple policy that greedily accepts the item with the highest reputation value will do well, as long as the reputation function is reasonably accurate. We use this greedy policy for Phase 2.

Our history-based reputation function R is simple: First, we find a list of persistent senders of legitimate mail from the same time period (we choose all senders that have appeared in at least 10 days), and for these IP addresses, we use their lifetime IP spam-ratio as their reputation value. For the remaining IP addresses, we use their cluster spam-ratio as their reputation value: for each week, we use the history of the preceding four weeks in computing the lifetime spam-ratio (defined over 4 weeks) for each cluster that sends mail. [5] In this way, we combine the results of the IP-based analysis and cluster-based analysis in Sec. 2 in designing the reputation function.

This reputation function is extremely simple, but it still illustrates the value of using a history-based reputation mechanism to tackle the mail server overload problem. We also note that the historical IP reputations based on network-aware clusters in this manner may not always be perfect predictors of spamming behaviour. While network-aware clusters are an aggregation technique with a basis in network structure, they could serve as a starting point for more complex clustering techniques, and these techniques may also incorporate finer notions of granularity and confidence.

A more sophisticated approach to using the history of IP addresses and network-aware clusters that addresses these concerns is likely to yield an improvement in performance, but is beyond the scope of this paper and left as future work. In the following section, we describe the performance benefits that we gain from using this reputation function in the evaluation.

4.3 Evaluation

We evaluate our history-based policies by replaying the traces of our data set on our simulator. Since the traces record each connection request with a time-stamp, we can replay the traces to simulate the exact workload received by the mail server. We do so, with the simplifying

---

[3]To ensure that the current mail server policy is not unfairly modelled under this simulation model, we evaluated greedy policies in another simulation model, in which each connection took z time units to process from start to end. The performance of the greedy policy was similar, therefore we do not describe the model further.

[4]Technically, this is slightly more complex: it examines if the load is below 75% of the server capacity allowed to Phase 1.

[5]One technical detail left to consider are the IP addresses originating from clusters without history. In our reputation function, any IP address that has no history-based reputation value is given a slightly bad reputation.

assumption that each incoming e-mail incurs the same computational cost. Since our traces are fixed, we simulate overload by decreasing the simulated server's capacity, and replaying the same traces. This way, we do not change the distribution and connection request times of IP addresses in the input traces between the different experiments. At the same time, it allows us to simulate, without changing the traces, how the mail server behaves as a function of the increasing workload.

Simulation Parameters: We now explain the parameters that we choose for our simulation. We choose the time t for the Phase 1 operation to be $4s$.[6] We use $60s$ for the timeout m, the waiting time in the queue before Phase 2 (it implies that mail will be delivered within 1 minute, or discarded after Phase 1). This appears to be sufficiently small so as to not noticeably affect the delivery of legitimate mail. [7]

To induce overload, we vary the capacity of the simulated mail server to 200, 100, 66, 50, and 40 messages/minute. The greedy policy processed an average of 95.2% of the messages received when the server capacity was set to 200 messages/minute, as seen in Table 2. At capacities larger than 200 messages/minute, the number of messages processed by the greedy policy grows very slowly, indicating that this is likely to be an effect of the distribution of connection requests in the traces. For this reason, we take capacity of 200/minute as the required server capacity. We then refer to the other server capacities in relation to required server capacity for this trace workload: a server with capacity of 100 messages/minute must process the same workload with half the capacity of the required server, so we define it to have an *overload-factor* of 2. Likewise, the server capacities we test $200, 100, 66, 50$ and 40 messages/minute have overload-factors of around $1, 2, 3, 4,$ and 5 respectively.

Recall that the parameter α is the cost of processing the message at Phase 1. We expect α to impact the performance, so we test two values $\alpha = 0.1, 0.5$ in the evaluation; recall that α is likely to be small, and so $\alpha = 0.5$ is a conservative choice here. The value of α has no effect on the performance of the greedy policy. For this reason, the discussion features only one greedy policy for all values of α. For the history-based policies, α sometimes has an effect on the performance, since these policies allow for a decision to be taken at Phase 2. We therefore refer to the history-based policies as 10-policy,

---

[6] We vary t for Phase 1 between 2-4s: our traces have a recorded time granularity of 1s, and the maximum seen in the traces before a disconnect was 4s. This does not appear to impact the results presented here, since both kinds of policies receive the same value of t. We present in the results for $t = 4s$

[7] This value also has no noticeable impact on our results when $m \geq 20s$ suggesting that most of the legitimate mail is processed quickly, or not at all.

and 50-policy, for $\alpha = 0.1$ and 0.5 respectively.

4.3.1 Impact on Legitimate mail

We first compare the number of legitimate mails accepted by the different policies over many time intervals, where each interval is an hour long. Since our goal is to maximize the amount of legitimate mail accepted, the primary metric we use is the *goodput ratio*: the ratio of legitimate mail accepted by the mail server to the total legitimate mail in the time interval. This is a natural metric to use, since it makes the different time intervals comparable, and so we can see if the policies are consistently better than the greedy policy, rather than being heavily weighted by the number of legitimate mails in a few time intervals. For the performance evaluation, we examine the average goodput ratio, the distribution of the goodput ratios and the goodput improvement factor.

Average Goodput Ratio: Table 1 shows the average goodput ratios for the different policies under different levels of overload. It shows that, on average, for each of these overloads, the goodput of any of the policies is better than the greedy policy. The difference is marginal at overload-factor 1, and increases quickly as the overload-factor increases: at overload-factor 4, the average goodput ratio is $64.3-64.5\%$ for any of the history-based policies, in comparison to 26.8% for the greedy policy. We also observe that the history-based policies scale more gracefully with the overload. Thus, we conclude that, on average, the history-based policies gain a significant improvement over the greedy policy.

Distribution of Goodput Ratios: While the average goodput ratio is a useful summarization tool, it does not give a complete picture of the performance. For this reason, we next compare the *distribution* of the server goodput in the different time intervals. Fig. 8(a)-(b) shows the CDF of the goodput ratios for the different policies, for two overload-factors: 1 and 4. We observe that the goodput ratio distributions are quite similar for the greedy and history-based policies when the overload-factor is 1 (Fig. 8(a)): about 60% of the time, all of the policies accept 100% messages. This changes drastically as the overload-factor increases. Fig. 8(b) shows the goodput ratio distributions for overload-factor 4. As much as 50% of the time, the greedy policy has a goodput-ratio of at most 0.25. By contrast, more than 90% of the time, the history-based policies have a goodput ratio of at least 0.5. The results show that the the history-based policies have a consistent and significant improvement over the greedy policy when the load is sufficiently high.

Improvement factor of Goodput-Ratios: Finally, we compare the goodput ratios on a per-interval basis. For this analysis, we focus on the 10-policy; our goal is to

see *how often* the 10-policy does better than the greedy algorithm. That is, for each time interval, we compute the *goodput-factor*, defined to be $\frac{\text{Goodput of 10-Policy}}{\text{Goodput of Greedy}}$. Fig. 8(c) plots how often goodput-factor lies between $90\% - 300\%$ for the different overload-factors. We note that when the overload-factor is 1, the performance impact of our history-based policy on the legitimate mail is marginal: in all the time intervals, the 10-policy has a goodput-factor of at least 90%, and over 95% of the time, it has a goodput factor of at least 99%. As the overload-factor increases, the amount of time intervals in which the 10-policy has a goodput-factor of 100% or more increases, meaning the number of time intervals in which the 10-policy does better than the greedy algorithm increases, as we would expect. When the overload-factor is 4, for example, 66% of the time, the goodput-factor is at least 200%: 10-policy accepts at least twice as many legitimate mails as the greedy algorithm. We conclude that in most time intervals, the history-based policies perform better than the greedy policy, and the factor of their improvement increases as the overload-factor increases.

Lastly, we note that the behaviour of the 10-policy and the 50-policy does not appear to differ too much when the overload-factor is sufficiently high or sufficiently low. With intermediate overload-factors, they perform slightly differently, as we see in Table 1: the 50-policy tends to be a little more conservative about accepting messages that may not have a good reputation in comparison to the 10-policy.

4.3.2 Impact on Throughput and Spam

While our primary metric of performance is the goodput, we are still interested in the impact of using the history-based policies on the total messages and spam processed by the mail server. While these are not our primary goals, they are still important since they give a picture of the complete effect of using these history-based policies.

Impact on Server Throughput: The history-based policies obviously gain their improvement by selectively choosing the IP addresses to process: it selectively accepts only good IP addresses in the incoming workload, if it is likely that the whole workload might not be processed. This may result in a decrease in server throughput in comparison to the greedy policy for certain load. For example, if the server receives a little less workload than it could process, the history-based policies may process fewer messages than the greedy policy, because they may reserve capacity for good IP addresses that they expect to see but which never actually appear. We observe this in our simulations and we discuss it now.
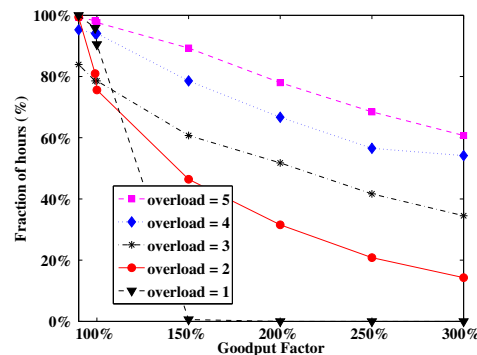
We define *throughput* to be fraction of the total messages processed by the server. Table 2 shows the average throughput achieved by both policies under vari-
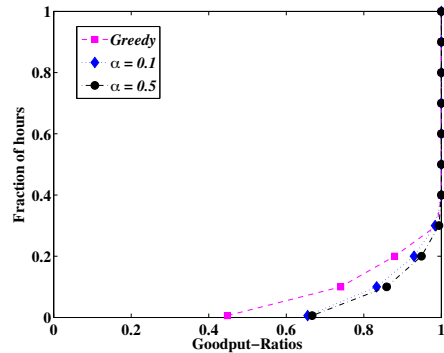
(a) Overload-Factor 1: CDF of goodput-ratios for all policies
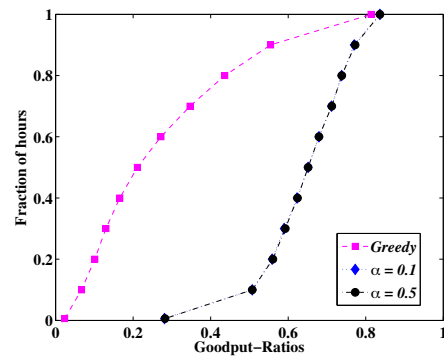
(b) Overload-Factor 4: CDF of goodput-ratios for all policies

(c) Goodput factor

Figure 8: (a) and (b): CDF of the goodput-ratios for two different overload-factors. (c): Performance improvement (goodput-factor) for the 10-policy for various overload factors

Overload Factor	Greedy	$\alpha = 0.1$	$\alpha = 0.5$
5	20.3	63	63.6
4	26.8	64.3	64.5
3	39.5	70.7	68.6
2	61.7	84.4	79.6
1	93.7	96	96.7

Table 1: Server Goodput (average, in %).

Overload Factor	Greedy	$\alpha = 0.1$	$\alpha = 0.5$
5	31.6	16.6	16.8
4	39.1	17	17
3	51.6	24.1	22.1
2	71.4	65.8	51.3
1	95.2	93.9	95

Table 2: Server throughput (average, in %).

Overload Factor	Greedy	$\alpha = 0.1$	$\alpha = 0.5$
5	32	14.8	14.9
4	39.5	15.1	15.1
3	52	20.1	15.2
2	71.7	65.2	50.2
1	95.2	93.8	94.9

Table 3: Spam accepted (average, in %).

ous capacities of the server. At overload-factor 1, when the greedy algorithm achieves an average throughput of 95%, the history-based policy algorithm achieves an average throughput of 93%. However, even at this point, the history-based policies accept a little more legitimate mail (on average) than the greedy policy. Note that by design, the history-based policies guarantee that when the server receives no more than 75% of its maximum load capacity, its performance is no different from normal.

Impact on Spam: We also explored the effect of the history-based policies on the number of spam messages accepted. Table 3 shows the average fraction of spam messages accepted by the policies under various overload factors. We see with an overload-factor of 1, the history-based policies accept only $0.3 - 1\%$ less spam than the greedy algorithm. As the overload-factor increases and the history-based policies grow more and more conservative in accepting suspected spam, the amount of spam accepted will decrease. For example, at a overload-factor of 2, this drops to $50.2\% - 65.5\%$ for the history-based policies. When the overload-factor increases to 4, the history-based policies accept less than 1/2 of the amount of spam accepted by the greedy policy. This suggests that if the server receives much more workload than it can process, the spam is affected much more than the legitimate mail. Therefore, the spammer would not have an incentive to increase the workload significantly, since it is the spam that gets most affected.

Thus, we have shown that our history-based policies achieve a significant and consistent performance improvement over the greedy policy when the server is under overload: we have seen this with multiple metrics of the goodput ratio. We have also seen that the history-based policies do not impact the performance of

the server too much when the server is *not* under overload. Finally, we have seen that the the spam is indeed affected when the server is significantly overloaded; this is precisely the behaviour we want to induce.

5 Related Work

Since spam is so pervasive, much effort has been expended in developing techniques that mitigate spam, and studies that understand various characteristics of spammers. In this section, we briefly survey some of the most related work. We first describe spam mitigation approaches and how they may relate to our work on the server overload problem. Then we discuss measurement studies that are related and complementary to our measurement work.

Traditionally, the two primary approaches to spam mitigation have used content-based spam-filtering and DNS blacklists. Content-based spam-filtering software [3, 1] is typically applied at the end of the mail processing queue, and there has been a lot of research [20, 17, 7, 16] in techniques for content-based analysis and understanding its limits. Agarwal et al. [6] propose content-based analysis to rate-limit spam at the router; this also reduces the load on the mail server, but is not useful for our situation as it may be too computationally expensive.

DNS blacklists [4, 5] are another popular way to reduce spam. Studies on DNS blacklists[14] have shown that over 90% of the spamming IP addresses were present in at least one blacklist at their time of appearance. Our approach is complementary to traditional blacklisting, and the more recent greylisting [13] techniques – we aim to prioritize the legitimate mail, and use the history of IP addresses to identify potential spammers.

Perhaps the closest in spirit to our work in mitigating server overload are those of Twining et al. [23] and Tang et al. [21]. Twining et al. describe a prioritization mechanism that delays spam more than it delays legitimate mail. However, their problem is different, as they eventually accept all email, but just delay the spam. Such an approach would not work when all the mail simply cannot be accepted. While Tang et al. [21] do not consider the problem of server overload, they describe a mechanism to assign trust to and classify IP addresses using SVMs. Our work differs in the way it gets the historical reputations – rather than using a blackbox learning algorithm, it uses the IP addresses and network-aware clusters, thus directly utilizing the structure of the network.

There has also been interest in using reputation mechanisms for identifying spam. There are a few commercial IP-based reputation systems (e.g., SenderBase [2], TrustedSource [22]). A general reputation system for internet defense has been proposed in [9]. There has

been work on using social network information for designing reputation-granting mechanisms to mitigate spam [10, 11, 8]. Prakash et al. [18] propose community-based filters trained with classifiers to identify spam. Our work differs from these reputation systems as it demonstrates the potential of using network-aware clusters to assign reputations to IP addresses for prioritizing legitimate mail.

Recently, there have been studies on characterizing spammers, legitimate senders and mail traffic, and we only discuss the most closely related work here. Ramachandran and Feamster [19] present a detailed analysis of the network-level characteristics of spammers. By contrast, our work focuses on the comparison between legitimate mail and spam and explores the stability of legitimate mail. We also use network-aware clusters to probabilistically distinguish the bulk of the legitimate mail from the spam. Gomes et al. [12] study the e-mail arrivals, size distributions and temporal locality that distinguish spam traffic from non-spam traffic; these are interesting features that distinguish spam and legitimate traffic patterns and provide general insights into behaviour. Our measurement study differs as it focuses on understanding the historical behaviour of mail servers at the network level that can be exploited to practical spam mitigation.

6 Conclusion

In this paper, we have focused on using IP addresses as a computationally-efficient tool for spam mitigation in situations when the distinction need not be perfectly accurate. We performed an extensive analysis of IP addresses and network-aware clusters to identify properties that can distinguish the bulk of the legitimate mail and spam. Our analysis of IP addresses indicated that the bulk of the legitimate mail comes from long-lived IP addresses, while the analysis of network-aware clusters indicated that the bulk of the spam comes from clusters that are relatively long-lived. With these insights, we proposed and simulated a history-based reputation mechanism for prioritizing legitimate mail when the mail server is overloaded. Our simulations show that the history and the structure of the IP addresses can be used to substantially reduce the adverse impact of mail server overload on legitimate mail, by up to a factor of 3.

7 Acknowledgements

References

[1] Brightmail. http://www.brightmail.com.

[2] SenderBase. http://www.senderbase.org.

[3] SpamAssassin. http://www.spamassassin.org.

[4] SpamCop. http://www.spamcop.net.

[5] SpamHaus. http://www.spamhaus.net.

[6] AGARWAL, B., KUMAR, N., AND MOLLE, M. Controlling spam e-mails at the routers. In *IEEE International Conference on Communications (ICC)* (2005).

[7] ANDROUTSOPOULOS, I., KOUTSIAS, J., CHANDRINOS, K., PALIOURAS, G., AND SPYROPOULOS, C. Spam filtering with Naive Bayes - which Naive Bayes? In *Third Conference on Email and Anti-Spam* (2006).

[8] BOYKIN, P. O., AND ROYCHOWDHURY, V. P. Leveraging social networks to fight spam. *Computer 38*, 4 (2005), 61–68.

[9] BRUMLEY, D., AND SONG, D. Towards attack-agnostic defenses. In *Proceedings of the First Workshop on Hot Topics in Security (HOTSEC)* (2006).

[10] GARISS, S., KAMISKY, M., FREEDMAN, M., KARP, B., MAZIERES, D., AND YU, H. Re: Reliable email. In *Proceedings of NSDI* (2006).

[11] GOLBECK, J., AND HENDLER, J. Reputation network analysis for e-mail filtering. In *First Conference on E-mail and Antispam* (2004).

[12] GOMES, L. H., CAZITA, C., ALMEIDA, J. M., ALMEIDA, V., AND WAGNER MEIRA, J. Characterizing a spam traffic. In *Proceedings of Internet Measurement Conference (IMC)* (2004).

[13] HARRIS, E. The next step in the spam control war: Greylisting. http://projects.puremagic.com/greylisting/.

[14] JUNG, J., AND SIT, E. An empirical study of spam traffic and the use of DNS black lists. In *Proceedings of Internet Measurement Conference (IMC)* (2004).

[15] KRISHNAMURTHY, B., AND WANG, J. On network-aware clustering of web clients. In *Proceedings of ACM SIGCOMM* (2000).

[16] LOWD, D., AND MEEK, C. Good word attacks on statistical spam filters. In *Second Conference on Email and Anti-Spam* (2005).

[17] MEDLOCK, B. An adaptive, semi-structured language model approach to spam filtering on a new corpus. In *Third Conference on Email and Anti-Spam* (2006).

[18] PRAKASH, V. V., AND O'DONNELL, A. Fighting spam with reputation systems. *Queue 3*, 9 (2005), 36–41.

[19] RAMACHANDRAN, A., AND FEAMSTER, N. Understanding the network-level behavior of spammers. In *Proceedings of ACM SIGCOMM* (2006).

[20] SAHAMI, M., DUMAIS, S., HECKERMAN, D., AND HORVITZ, E. A Bayesian approach to filtering junk E-mail. In *Learning for Text Categorization: Papers from the 1998 Workshop* (1998), AAAI Technical Report WS-98-05.

[21] TANG, Y., KRASSER, S., AND JUDGE, P. Fast and effective spam sender detection with granular SVM on highly imbalanced server behavior data. In *2nd International Conference on Collaborative Computing: Networking, Applications and Worksharing* (2006).

[22] TRUSTEDSOURCE. http://www.trustedsource.org.

[23] TWINING, D., WILLIAMSON, M. M., MOWBRAY, M., AND RAHMOUNI, M. Email prioritization: Reducing delays on legitimate mail caused by junk mail. In *USENIX Annual Technical Conference* (2004).

A Appendix

We present here the details of the policy used in Phase 1. for the history-based policies. In detail, the policy is the following: If the load is less than 75% of its capacity, the policy accepts all SMTP connections requests, regardless of the reputation of the IP address. If the load is greater than 75% of the capacity, the policy starts considering the reputation of the IP address and the legitimate mail that it expects to have to process in the near future.

For this purpose, it uses a distribution of the number of emails expected in the next t time units from reputation value at most k (for multiple k values), that is calculated based on the history of the distribution of mail arrival. Since our reputation function is the lifetime spam-ratio, a low reputation value is a good reputation, and a high reputation value is a bad reputation. Then it does the following: (a) given the current load, it computes the smallest k' such that all expected mail with reputations with $k \leq k'$ can be processed on the server (b) it looks up the reputation of the IP address, and checks if it is higher than k'. (If the IP address does not have a known reputation value, and it does not belong to a cluster with a known reputation, then the IP address is assigned a relatively higher k' value. If $k' \leq k$, then the connection request of IP address is accepted, otherwise, it is rejected.