DMTP: Controlling Spam Through Message Delivery Differentiation

Zhenhai Duan
Computer Science Department
Florida State University
Tallahassee, FL 32306
Email: {duan}@cs.fsu.edu

Yingfei Dong
Dept. of Electrical Engineering
University of Hawaii
Honolulu, HI 96822
Email: {yingfei}@hawaii.edu

Kartik Gopalan
Computer Science Department
Florida State University
Tallahassee, FL 32306
Email: {kartik}@cs.fsu.edu

Abstract

Unsolicited commercial email, commonly known as spam, has become a pressing problem in today's Internet. In this paper we re-examine the architectural foundations of the current email delivery system that are responsible for the proliferation of email spam. We argue that the difficulties in controlling spam stem from the fact that the current email system is fundamentally sender-driven and distinctly lacks receiver control over email delivery. Based on these observations we propose a Differentiated Mail Transfer Protocol (DMTP), which grants receivers greater control over how messages from different senders should be delivered on the Internet. In addition, we also develop a formal mathematical model to study the effectiveness of DMTP in controlling spam. Through numerical experiments we demonstrate that DMTP can effectively reduce the maximum revenue that a spammer can gather. Moreover, compared to the current SMTP-based email system, the proposed email system can force spammers to stay online for longer periods of time, which may significantly improve the performance of various real-time blacklists of spammers. In addition, DMTP provides an incremental deployment path from the current SMTP-based system in today's Internet.

I. Introduction

Unsolicited commercial email, commonly known as *spam*, is a pressing problem on the Internet. In addition to undermining the usability of the current email system, spam also costs industry billions of dollars each year in recent years [9], [26]. In response, the networking research and industrial communities have proposed a large number of anti-spam countermeasures, including numerous email spam filters [3], [6], [12], [13], [22], [23], [25], sender authentication schemes [8], [18], [20], and sender-discouragement mechanisms (to increase the cost of sending email such as paid email) [11], [16]. Some of the schemes have been deployed on the Internet. On the other hand, despite these anti-spam efforts, in recent times the proportion of email spam seen on the Internet has been continuously on the rise [4], [6].

A. Why Is It so Hard to Control Spam?

The current email system uses the Simple Mail Transfer Protocol (SMTP) to deliver messages from sender to receiver [17]. While simple, such a system also provides an ideal platform for spammers to act as parasites. It is our contention that, in order to effectively control spam, we must design and deploy an email delivery system that can proactively resist spam in the first place. As a first step toward this goal, in this paper we examine the architectural aspects of the current email system that are responsible for the proliferation of spam and propose a Differentiated Mail Transfer Protocol (DMTP) that aims to overcome these limitations based on the following three key insights.

Moving to a receiver-driven model: First, the current email system is fundamentally sender-driven and distinctly lacks receiver control over the message delivery mechanism. For example, in the current SMTP-based email system, any user can send an email to another at will, regardless of whether or not the receiver is willing to accept the message. In the early days of the Internet development, this was not a big problem as people on the network largely trusted each other. However, since the commercialization of the Internet in the mid-1990s, the nature of the Internet community has changed. It has become less trustworthy, and the emergence of email spam is one of the most notable examples of this change. In order to effectively address the issue of spam in the untrustworthy Internet, we argue that *receivers must gain greater control over if and when a message should be delivered to them*.

Eliminating economy of scale: Secondly, volume is the most crucial factor in making email spam a profitable business. In order to squeeze spammers out of business, we must eradicate the economy of scale they rely on. However, in the current email system, the sending rate of spam is, to a large extent, only constrained by the processing power and network connectivity of spammers' own mail servers, of which the spammers have complete control. Nowadays, with increasingly-powerful (and cheaper) PCs and ubiquitous high-speed Internet access, spammers can push out a deluge of spam within a very short period of time, making spamming profitable because of the economy of scale. We contend that *the sending rate of spam must be regulated, ideally under the control of email receivers*, in order to retain spam.

Increasing accountability: Lastly, the current email system makes it hard to hold spammers accountable for spamming. Spammers can vanish (go offline) immediately after pushing a deluge of spam to receivers (recall that this can be done within a very short period of time). This makes it quick and easy for spammers to hide their identities and provides spammers with the flexibility to frequently change their locations and/or Internet service providers—complicating the effort to filter spam based on the IP addresses of sender mail servers, such as various real-time blacklists (RBLs) [22]. We argue that in order to hold spammers accountable and to make RBLs more effective, *we must force spammers to stay online for longer periods of time*.

B. Contributions of this Paper

Based on these observations we propose a Differentiated Mail Transfer Protocol (DMTP) as a countermeasure to the spam problem. A key feature of DMTP is that it grants receivers greater control over the message delivery mechanism. In DMTP, a receiver can classify senders into different classes and treat the delivery of messages from each class differently. For example, although regular contacts of a receiver can directly send messages to the receiver, unknown senders need to store messages in the *senders' own mail servers*. Such messages are only retrieved by the receiver *if and when* he wishes to do so.

DMTP provides us with several important advantages in controlling spam: 1) the delivery rate of spam is determined by the spam retrieval behavior of receivers instead of being controlled by spammers; 2) spammers are forced to stay online for longer periods of time (because the sending rate of spam is regulated by the spam retrieval rate of receivers), which can significantly improve the performance of RBLs; 3) regular correspondents of a receiver do not need to make any extra effort to communicate with the receiver—correspondence from regular contacts is handled in the same manner as in the current SMTP-based email system; 4) DMTP can be easily deployed on the Internet incrementally.

In this paper we present the design of DMTP and formally model its effectiveness in controlling spam. Through numerical analyses we show that DMTP can significantly reduce the maximum revenue that a spammer can obtain. In addition, a spammer has to stay online for a much longer period of time in order to obtain the maximum revenue.

The remainder of the paper is organized as follows. In Section II we re-examine two common traffic delivery models on the Internet: sender push vs. receiver pull, and discuss their implications on controlling spam. In Section III we present the design of DMTP, which employs a variant of the receiver-pull model. We formally model the effectiveness of DMTP in controlling spam and perform numerical analyses in Section IV. In Section V we discuss practical deployment issues of DMTP on the Internet. After describing related work in Section VI, we conclude the paper and outline our ongoing work in Section VII.

II. Push vs. Pull: Implications of Protocol Design Choice

Asynchronous messages like email are delivered on the Internet primarily using two different models: *sender-push and receiver-pull* (or a combination of the two). In this section we discuss the implications of the two models on controlling unwanted traffic on the Internet and illustrate that the receiver-pull model has several important advantages in discouraging unwanted Internet traffic such as email spam. In light of these advantages, in the next section we develop a new email delivery protocol based on the receiver-pull model.

The two models differ in who initiates the message delivery process. In the sender-push model, senders control the delivery of traffic, and receivers passively accept whatever the senders push to them. The current SMTP-based email delivery system is a typical example of this model. In contrast, the receiver-pull model grants receivers the control over if and when they want to retrieve data from the senders. In this model, senders can only prepare the

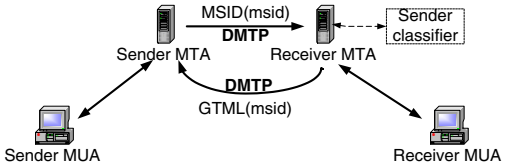

Fig. 1. Illustration of DTMP-based email system.

TABLE I
NEWS COMMANDS/REPLY CODE DEFINED IN DMTP.

Commands/Replies	Explanation
`MSID`	For SMTA to inform RMTA the *msid* of a message
`GTML`	For RMTA to retrieve a message from SMTA
`253`	For RMTA to inform SMTA to send *msid* (`MSID`) instead of messages (`DATA`)

data but they cannot push the data to receivers. Examples of the receiver-pull model include the HTTP-based web access services and the FTP-based file transfers.

While both simple and convenient, the sender-push model has a big disadvantage in controlling unwanted Internet traffic: in this model it is senders who completely control *what* messages are delivered and *when* the messages are delivered. Receivers have neither the knowledge of what messages they will receive, nor when the messages will be received. Receivers are ideally expected to receive the entire messages before processing or discarding the messages. Moreover, senders can vanish immediately after the messages are pushed out. By contrast, the receiver-pull model comes with several appealing advantages because it grants receivers greater control over the message delivery mechanism. It takes advantage of the fact that receivers have more reliable knowledge of what traffic they want to receive. In this model, receivers have the freedom to first determine the reputation of the senders (and their own level of interest in the contents) *before* they actually request the content. Moreover, it becomes the responsibility of senders to store and manage the messages till the receivers are ready to retrieve the messages. This forces malicious senders to stay online and reveal their identities for larger windows of time.

A legitimate concern with the receiver-pull model is that it may increase the cost of sending messages for malicious as well as legitimate senders. We show in the next section that, using simple design optimizations, we can easily lower the sending cost for legitimate senders while still retaining the benefits of the receiver-pull model. In summary, although the receiver-pull model may result in slightly greater protocol complexity, it can greatly help to simplify the control of unwanted traffic such as spam on the Internet, and should be considered early during any communication system design.

III. DMTP: A DIFFERENTIATED MAIL TRANSFER PROTOCOL

DMTP is designed based on a variant of the receiver-pull model, where senders are allowed to first express an intent to send message to a receiver via a small intention message. If the receiver happens to be interested, he contacts the sender and retrieves the content message. Figure 1 illustrates the basic architecture of the new email delivery system. Before we delve into the details of DMTP, it is worth noting that the new system extends the current SMTP protocol [17] by adding two new commands—`MSID` and `GTML`, and one new reply code—253 (see Table I). All the commands and reply codes in SMTP are also supported in the new system. We explain the new commands and reply code when we use them.

A. Differentiating Message Deliveries

As discussed in the last section the receiver-pull model increases the cost of sending messages for both malicious and legitimate senders. To address this issue DMTP is designed to support a hybrid email delivery system where both the sender-push and receiver-pull models can be employed. Specifically, each receiver can classify email senders into three disjoint classes and treat the delivery of messages from each of them differently: 1) *well-known spammers*, whose messages will be directly rejected; 2) *regular contacts*, whose messages can be directly pushed

```
Require: SPC: well-known spammer class;
Require: RCC: regular contact class;
Receiving TCP session open request on port 25;
ip = Get IP address of sender mail server;
if (ip ∈ SPC) then
    /* well-known spammers */
    reply with 550 (to decline TCP session opening request);
    close TCP session;
else if (ip ∈ RCC) then
    /* regular contacts */
    reply with 220 (to accept TCP session opening request);
    proceed as if SMTP used;
else
    /* unclassified senders */
    reply with 253 (see Table I);
    accept MSID command;
    reject DATA command;
end if
```

Fig. 2. Algorithm for receivers to handle message delivery requests in DMTP.

from the senders to the receiver using the current SMTP protocol; and 3) *unclassified senders*—senders that are neither well-known spammers nor regular contacts. Unlike regular contacts, unclassified senders cannot directly push a message in its entirety to the receiver. Such messages need to be stored and managed by the *senders' mail servers*, and only the envelope of the messages can be directly delivered to the receiver to notify the pending messages.

Senders can be defined at the granularity of email addresses as well as IP addresses (and domain names) of sender mail servers. Given that it is easy to fake email addresses in the current Internet, we envision that sender classification will be performed at the granularity of IP addresses when DMTP is first deployed.

Fig. 2 summarizes the algorithm of handling message delivery requests at a DMTP receiver. In the figure we have assumed that the sender classification is only supported at the IP addresses (and domain names) level. Sender classification defined at email address level can be easily incorporated into the algorithm. In the rest of this section we focus on the handling of messages from unclassified senders. The handling of messages from well-know spammers and regular contacts is the same as in the current practice [17], [22], and we omit the description.

B. Unclassified Sender: Message Composition and Receiver Notification

Like in the current email system, an (unclassified) sender uses a Mail User Agent (MUA) to compose outgoing messages [17]. After a message is composed by the sender, the sender delivers the message to the sender Mail Transfer Agent (MTA). For simplicity, we refer to a sender MTA server as an SMTA, and a receiver MTA server as an RMTA.

All the outgoing messages of unclassified senders are stored at the SMTAs. For this purpose, an SMTA maintains an outgoing message folder for each *sender*. Instead of a complete message being directly pushed from the SMTA to the RMTA, only the envelope of the message is delivered. In particular, the SMTA notifies the RMTA about the pending message via the new *message identifier* command `MSID` (see Table I), which contains the unique identifier *msid* of the message. The *msid* is used by the receiver to retrieve the corresponding message.[1] The identifier of a message is generated based on the sender, the receiver, and the message.

[1]Note the fundamental difference between message pull in the new email system and URL embedded in many current spam messages. The address in the URL is normally not related to the sending machine of the message. In contrast, outgoing messages in the new email system have to be stored on the sender mail servers.

TABLE II

NOTATIONS USED IN THE SPAMMER REVENUE MODEL.

Notation	Explanation	Setting
N	Number of email addresses maintained by spammer	10M
x	Number of machines used by spammer	62
k	Sending speed of a machine (messages/unit time)	100K
y	Cost paid by spammer per machine per unit time	0.1
g	Gains of spammer for each message delivered	0.005
p	Probability that a receiver reports a spamming machine	0.001
q	Number of reports required for RBL to blacklist a machine	50
r	Mean spam retrieval rate of receivers (retrievals/unit time)	2500

C. Receiver: Pulling Messages from Unclassified Senders

The new email delivery system grants greater control to receivers regarding if and when receivers want to read a message; senders cannot arbitrarily push a message to them. Receivers can be discriminate about which messages need to be retrieved, and which ones need not. If a receiver indeed wants to read a message, he will inform his own RMTA, and the RMTA will retrieve the message from the SMTA on behalf of the receiver. An RMTA retrieves an email message using the new *get mail* command `GTML` (see Table I), which includes the identifier *msid* of the message to be retrieved. After the message has been pulled to the RMTA, conventional virus/worm scanning tools and content-based spam filters can be applied to further alert the receiver about potential virus or spam. Therefore, the new email system does not exclude the use of existing email protection schemes. For security reasons, when an SMTA receives the `GTML` command, it needs to verify that the corresponding message is for the corresponding email receiver, and the requesting MTA is the mail server responsible for the receiver.

D. Minimizing the Impact of Intent Messages

It is conceivable that before the majority of spammers are squeezed out of business, a large number of small intent messages may be delivered to Internet email users when DMTP is first deployed on the Internet. A legitimate concern is that email users may be overwhelmed by such small intent messages. This problem can be alleviated by, e.g., quarantining intent messages: RMTA will only deliver messages from regular contacts to receivers immediately; all the intent messages from unclassified senders will be first quarantined at the RMTA and only delivered to the receivers periodically in a single digest message. The interval over which the RMTA delivers the digest email of intent messages to a receiver can be configured by the receiver. A similar idea has been supported in commercial products and employed in real-world systems to handle spam messages [24], [15]. As more spammers run out of business because of the increased adoption of DMTP, intent messages related to spamming will decrease and be less of a concern. (Rather, they are used for legitimate reasons for first-time correspondents to communicate.)

IV. PERFORMANCE EVALUATION

In this section we first develop a simple mathematical model to investigate the revenue that a spammer can gather by spamming a message to a set of Internet users. Based on this model, we then perform numerical experiments to study the effectiveness of DMTP in controlling spam, and how the behaviors of both spammers and receivers affect the spammers' revenue.

A. A Simple Model of Spammer Revenue

Table II summarizes the notations used in this section. Consider a spammer s. We assume that s maintains a set of N email addresses to which he can send spam emails. In this model we establish the expected revenue the spammer can gather by sending a single message to the N email addresses. We assume the spammer owns or rents x machines to send spam (each with a unique IP address). On average, each machine is capable of sending k messages per unit time (which is only constrained by the processing power and Internet access speed of the machines). The spamming task is equally partitioned over the x machines, that is, each machine needs to send the message to N/x receivers. For each machine, the spammer needs to pay y units of cost for each unit of time (e.g.,

for Internet access or renting machines from hackers or time spent in recruiting zombies). In return, the spammer obtains g units of gain for each message delivered.

For simplicity, we assume there is a central real-time blacklist of well-known spammers, which is used by all receivers. Before the spammer starts spamming, we assume that none of the x machines managed by the spammer is listed by the central RBL. Instead, they are in the unclassified-sender class of all N receivers. (Sender classification is defined at the granularity of IP addresses.) When a receiver retrieves a message from the spammer, it will report the IP address of the corresponding SMTA to the central RBL with a probability of p. (We assume that intent messages are directly delivered to end users instead of first being quarantined at the RMTAs.) Furthermore, the central RBL requires at least q reports of a spamming machine before adding the corresponding IP address into its blacklist. After an IP address is added to the blacklist, the spammer can no longer send messages from the corresponding SMTA. To simplify, we assume that the spammer has the precise knowledge of the time when an SMTA is blacklisted and will disconnect the machine to minimize its own cost.

We assume the arrivals of spam retrievals from receivers follow a Poisson distribution, with a mean arrival (i.e., retrieval) rate r (retrievals per unit time). Given that the list of email addresses maintained by a spammer is in general large, we assume the spam retrieval rate r is a constant over time. Below we derive the expected revenue $U(t)$ of the spammer at time t, assuming the time for the spammer to start spamming the message to the N receivers is zero.

Let $R(t)$ denote the expected number of receivers who have retrieved the message at time t. It is not too hard to see that $R(t) = \min\{rt, xq/p\}$. Let $f(t)$ denote the expected number of messages delivered by the spammer at time t (across all x machines), we have $f(t) = \min\{N, xkt, R(t)\}$. Consequently, the expected income of the spammer at time t is $gf(t)$. On average, it takes N/r units of time for the spammers to deliver the message to all receivers, and it takes $(q/p)/(r/x)$ units of time for the central RBL to blacklist an SMTA (assuming $r \ll k$ and all x machines are accessed with the same probability). Therefore, the total expected cost $c(t)$ paid by the spammer at time t is $c(t) = xy \min\{t, N/r, (q/p)/(r/x)\}$. Hence, in the DMTP-based email system the total expected revenue of the spammer at time t is

$$U_{DMTP}(t) = gf(t) - c(t) = g \min\{N, xkt, R(t)\} - xy \min\{t, N/r, (q/p)/(r/x)\}. \quad (1)$$

We can similarly derive the total expected revenue of the spammer at time t in the current SMTP-based email system, which is given below

$$U_{SMTP}(t) = g \min\{N, xkt\} - xy \min\{t, (N/x)/k\}. \quad (2)$$

In the above equation, we have assumed that k is large enough that the spammer can finish sending the message to all receivers before the SMTAs are blacklisted.

Comparing Eq. (1) and Eq. (2), we see that while the revenue of the spammer is largely determined by the *sending* speed of its SMTAs in the current SMTP-based email system, in the DMTP-based email system its ability to spam is greatly constrained by the message retrieval behavior of the receivers. The slower the receivers are in retrieving the message, the longer the spammer needs to stay online; the higher the probability is for receivers to report spamming SMTAs to the central RBL, the earlier the spamming SMTAs are blacklisted.

B. Numerical Studies

In this section we perform numerical experiments to study the effectiveness of the proposed DMTP protocol in controlling spam using the model developed in the last subsection. We also investigate how the behaviors of both spammers and receivers affect the spammers' revenue. Table II (third column) presents the parameter values we used in the numerical studies, unless otherwise stated.

First, we study how the proposed DMTP protocol helps to reduce the maximum revenue of a spammer (by spamming a message to N receivers) and forces the spammer to stay online (to improve the performance of RBLs). Fig. 3 shows the revenues of the spammer as time evolves in both the current SMTP-based email system (curved marked as *Without DMTP*) and the proposed DMTP-based email system. From the figure we see that, in the current email system, the spammer can gather the maximum revenue (49990) within 2 units of time. This means that the spammer can quickly push out the message to all the receivers and then vanish, long before any RBLs

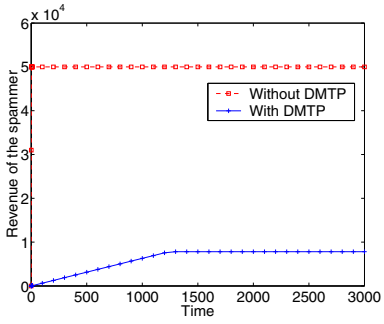

Fig. 3. Expected spammer revenue.

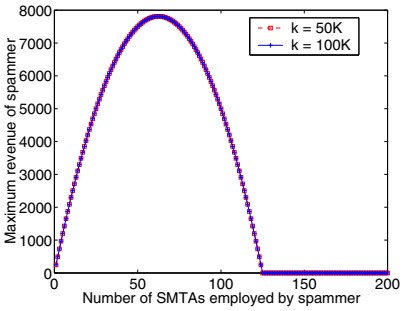

Fig. 4. Impact of number of SMTAs.

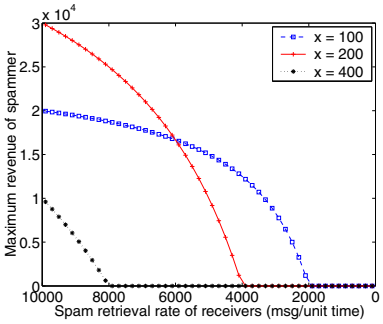

Fig. 5. Impact of spam retrieval rate.

can identify it. In contrast, in the DMTP-based email system, the maximum revenue is 7812 units, only about 16% of the spammer maximum revenue in the current email system. Moreover, in order for the spammer to gather the maximum revenue, the spammer has to stay online for a much longer time window (1240 units of time). This can significantly improve the performance of RBLs. Note also that the revenues will not decrease once they reach the maximum values. This is because a spammer disconnects an SMTA to minimize the cost once the SMTA has finished sending the message to all receivers (in SMTP) or it is blacklisted (in DMTP).

Next, we investigate the impact of the number of SMTAs employed by a spammer on the maximum spammer revenue in the DMTP-based email system. Fig. 4 shows the *maximum* spammer revenue as a function of the number of SMTAs employed by the spammer for $k = 50K$ and $100K$, respectively. Note first that increasing the sending speed of spam from $k = 50K$ to $100K$ will not result in a higher maximum spammer revenue. Indeed, after the spam sending speed exceeds the spam retrieval rate of receivers, it will not affect the maximum spammer revenue. Now let us examine how the number of SMTAs employed by a spammer will affect the maximum spammer revenue. As we can see that the spammer has some initial gains by increasing the number of SMTAs (when the number is less than 62). This is because as the number of SMTAs increases, it takes a longer time for all the SMTAs to be blacklisted by the central RBL, and the message can be retrieved by more receivers. Fortunately, the spammer cannot indefinitely increase the number of SMTAs to evade RBLs. When the spammer employs more than 62 SMTAs, his maximum revenue actually starts to drop, as the income of delivering the message to new receivers can no longer recompense the cost to deploy the new SMTAs.

In the last set of numerical experiments, we study the effects of the spam retrieval rate of receivers on the maximum spammer revenue. Fig. 5 depicts the *maximum* spammer revenue as a function of the spam retrieval rate of receivers for number of SMTAs $x = 100, 200, 400$, respectively. As we can see from the figure, the maximum spammer revenue decreases as the receivers reduce their retrieval rate of messages from the unclassified SMTAs for all three cases. Moreover, when the retrieval rate is sufficiently low (for example, less than 2000 retrievals per unit time when $x = 100$), the spammer cannot gather any revenue from spamming. More importantly, when a spammer recruits more SMTAs to send spam, it requires a larger threshold of spam retrieval rates for the spammer to gather any revenue (for example, 4000 when $x = 200$, compared to 2000 when $x = 100$). This again demonstrates that spammers cannot gather more revenue by indefinitely recruiting more SMTAs. As more spammers run out of business because of the increased adoption of DMTP, the email spam problem will be effectively controlled on the Internet.

V. Practical Deployment Issues

A. Incremental Deployment

DMTP can be easily deployed on the Internet incrementally. The basic idea is to combine DMTP with a sender-discouragement scheme (such as asking senders to solve a puzzle). However, unlike existing sender-discouragement schemes, we only require senders in the unclassified-sender class to make the extra effort in sending a message. In this section we outline one such approach. It is worth noting, however, that DMTP can be incrementally deployed on the Internet in other fashions. In the following, we assume that the RMTA in consideration supports the DMTP

protocol, and show how it interacts with the rest of the world. For simplicity we assume that the sender classification is performed at the granularity of IP addresses (or domain names) of SMTAs.

In order to support incremental deployment, RMTA supporting DMTP needs to know if the SMTA also supports DMTP. For this purpose, an SMTA supporting DMTP will inform the RMTA this fact by including keyword "DMTP" in the greeting command `EHLO` (or `HELO`). Figure 6 presents the algorithm used by receivers to handle message delivery requests in supporting incremental deployment of DMTP.

```
Require: SPC: well-known spammer class;
Require: RCC: regular contact class;
Receiving TCP session open request on port 25;
ip = Get IP address of sender mail server;
if (ip ∈ SPC) then
    /* well-known spammers */
    reply with 550 (to decline TCP session opening request);
    close TCP session;
else if (ip ∈ RCC) then
    /* regular contacts */
    reply with 220 (to accept TCP session opening request);
    proceed as if SMTP used;
else
    /* unclassified senders */
    reply with 220 (to accept TCP session opening request);
    proceed to the EHLO command;
    if (found keyword "DMTP" in the EHLO command) then
        /* sender supports DMTP */
        proceed according to DMTP;
    else
        /* sender does not support DMTP */
        respond to DATA command with 354;
        receive message;
        respond with 550 (permanent error);
        store message, send puzzle;
        message invisible to user;
        /* message becomes visible to user only after puzzle solved */
    end if
end if
```

Fig. 6. Handling message delivery requests at RMTAs for incremental deployment of DMTP.

B. Other Issues

Security of message retrieval: A potential concern with the receiver-pull model is security. However as we discuss below, the potential security issue arising from this model is no worse than the current SMTP model. First, important messages are normally communicated amongst regular contacts, which are handled in DMTP in the same way as in the current email system. Secondly, individual users cannot retrieve messages from a remote SMTA directly, they rely on their corresponding RMTAs to retrieve messages (from unclassified senders). Lastly, *msid*s are generated randomly based on the messages (and senders and receivers); they cannot be easily guessed.

Mailing list: We believe that in the future all mailing lists will be mediated and content-based spam filters will be universally deployed by all mailing lists. In DMTP, we suggest all users to add their mailing lists into their *regular contacts*. In this way, the RMTA of a user can directly accept the message from the mailing list, without

putting any extra burden on the MTA of a mailing list and mediator. Similarly, the MTA of a mailing list should also add all members into its *regular contacts*, such that it can directly receive messages from its members.

Electronic greeting card delivery services: This type of services puts great challenges on sender authentication. Sender Rewriting Scheme (SRS) [28] was proposed to mitigate this issue. A main challenge for DMTP is how to handle the delivery of messages whose sender addresses have been rewritten by SRS. One possible approach is to let MTAs maintain the reputations of the E-Card sites, and only allow sites with good reputation to directly deliver a message to the RMTA. For other sites, only the headers are delivered. End users need to contact the original senders before the complete message is retrieved from the E-Card sites.

Populating regular contacts classes: It is conceivable that a receiver may want to communicate with someone who is currently not in the regular contact list. In the following we outline an out-of-band approach: such senders need to send messages through a web-based interface, and corresponding mechanisms are called for to ensure that automatic email agents cannot fill the web forms and send messages. The RMTA will directly accept the complete messages and mark them as *OUTOFBAND*. After such friends have been added into the regular contacts class, they do not need to take efforts to register again.

Exporting user regular contacts to service providers: Users may not be willing to export their own regular contact lists (especially the ones at the email address level) to the service providers. Some secure mechanisms to conceal the exact identifications of users' regular contacts can be used, such as Bloom filters [2]. Users hash their regular contacts to a bloom filter and export the bloom filter to the corresponding RMTA instead of the exact regular contacts. The RMTA relies on the bloom filter to detect if a sender is in the user's regular contact list (note that bloom filters may incur some false positives).

User-perceived system performance: Given the ever-increasing network speeds, we do not expect any degradation of user-perceived email reading experience, although some messages–the ones from unclassified senders–need to be retrieved from a remote mail server. We plan to formally study this issue in our future work (but note the largely satisfactory web-surfing experience, where, in a similar manner, a web page needs to be remotely fetched).

VI. Related Work

The most widely deployed anti-spam solutions today are reactive content filters that scan the contents of the message at the receiver's MTA after the message has been delivered. However, none of them can achieve 100% accuracy, and spammers quickly adapt to counter the strategies used by these filters. In addition, content filtering will no longer serve as long-term viable solution once email messages begin to be encrypted using receivers' public keys [21]. Instead, we have advocated fundamental changes in protocol-level design to a pull-based model.

Like DMTP, FairUCE [5] also advocates the usage of sender classifiers. However, it is still a push-based model in which network reputation, along with receiver defined whitelist and blacklist, is used to determine whether to accept a message. IM2000 [1] also advocates a pull-based model like DMTP. However, unlike DMTP, all outgoing messages need to be stored at sender MTAs and receivers need to retrieve all the messages remotely, regardless of where the messages come from. In addition, IM2000 is not incrementally deployable and requires massive infrastructure changes. Li *et al* proposed a method to slow down spam delivery by damping the corresponding TCP sessions [19]. However, the long-term impact of modifying the behavior of TCP for a specific application is not clear, and spammers may respond by changing sender MTA's TCP behavior. In the Greylisting [14] approach, a message from a new sender is temporarily rejected upon the first delivery attempt, the underlying assumption being that spammers will not re-send a message whereas regular MTAs will. However, it is only a matter of time before spammers adapt to this technique by re-sending their message. Sender authentication schemes such as [8], [18] can help improve the accountability of email senders. However, they cannot control the delivery of spam by themselves.

The Internet Message Access Protocol (IMAP) allows a user to retrieve part of a message, such as the message header without fetching the complete message, from *his mail server* [7]. However, it works only between the user's MUA and his local mail server. The complete message is first delivered from the sender MTA to the receiver MTA. Email Prioritization was proposed in [27] as a way to control the impact of spam on legitimate messages. However, the performance of the system depends on how well it can predict that an incoming message is spam. Moreover, spammers still have the incentive to send a large number of messages given that the entire messages including

both headers and bodies are still delivered from the sender to the receiver (even though they may do so at the cost of purchasing more machines). Gburzynski and Maitan proposed to use Email aliases to fight Email spam [10], where different Email aliases can be created for different purposes and used over a specific duration. However, its effectiveness relies on hiding Email addresses and their aliases. Moreover, users have more burdens to manage their accounts. For example, they need to create Email aliases and disseminate them to intended correspondents.

VII. Conclusion and Ongoing Work

In this paper we examined the architectural aspects of the current email system that are responsible for the proliferation of spam, and proposed a Differentiated Mail Transfer Protocol to control spam. In addition, we also developed a formal model to study the performance of DMTP. Through numerical experiments we demonstrated that DMTP can significantly reduce the maximum spammer revenue. Moreover, it also forces spammers to stay online for longer periods of time, which helps improve the performance of real-time blacklists of spammers. Currently we are developing a prototype of DMTP. We plan to further investigate the performance of DMTP based on the prototype and simulations.

References

[1] D. Bernstein. Internet mail 2000 (IM2000). http://cr.yp.to/im2000.html.

[2] A. Broder and M. Mitzenmacher. Network applications of bloom filters: A survey. In *Proceedings of the 40th Annual Allerton Conference on Communication, Control, and Computing*, 2002.

[3] X. Carreras and L. Márquez. Boosting trees for anti-spam email filtering. In *Proceedings of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing*, Tzigov Chark, BG, 2001.

[4] T. Claburn. Big guns aim at spam. *Information Week*, March 2004.

[5] IBM Corporation. Fair use of unsolicited commercial email FairUCE, November 2004. http://www.alphaworks.ibm.com/tech/fairuce.

[6] L. Cranor and B. Lamacchia. Spam! *Communications of the ACM*, 41:74–83, August 1998.

[7] M. Crispin. Internet message access protocol - version 4rev1. *IETF RFC3501*, March 2003.

[8] M. Delany. Domain-based email authentication using public-keys avertised in the DNS (domainkeys). Internet Draft, August 2004. Work in Progress.

[9] Ferris Research. Spam control research reports. http://www.ferris.com/.

[10] P. Gburzynski and J. Maitan. Fighting the spam wars: A remailer approach with restrictive aliasing. *ACM Transactions on Internet Technology*, 4(1):1–30, February 2004.

[11] J. Goodman and R. Rounthwaite. Stopping outgoing spam. In *Proc. of EC'04*, 2004.

[12] P. Graham. Better bayesian filtering. *http://www.paulgraham.com/better.html*, January 2003.

[13] P. Graham. A plan for spam. *http://www.paulgraham.com/spam.html*, January 2003.

[14] E. Harris. The next step in the spam control war: Greylisting. White Paper, August 2003.

[15] ITS. Spam at the university of hawaii. http://www.hawaii.edu/infotech/spam/spam.html. Last checked: 11/19/2005.

[16] A. Juels and J. Brainard. Client puzzles: A cryptographic defense against connection depletion attacks. In *Proceedings of NDSS-1999*, February 1999.

[17] J. Klensin. Simple mail transfer protocol. RFC 2821, April 2001.

[18] M. Lentczner and M. W. Wong. Sender policy framework (spf): Authorizing use of domains in MAIL FROM. Internet Draft, October 2004. Work in Progress.

[19] K. Li, C. Pu, and M. Ahamad. Resisting spam delivery by TCP damping. In *Proceedings of First Conference on Email and Anti-Spam (CEAS)*, July 2004.

[20] J. Lyon and M. Wong. Sender ID: Authenticating e-mail. Internet Draft, August 2004. Work in Progress.

[21] P. Mannion. Interview: Ethernet's inventor sounds off. *Information Week*, November 2005.

[22] RBL. Real-time spam black lists (rbl). http://www.email-policy.com/Spam-black-lists.htm.

[23] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk E-mail. In *Learning for Text Categorization: Papers from the 1998 Workshop*, Madison, Wisconsin, 1998. AAAI Technical Report WS-98-05.

[24] Sophos Plc. Sophos plc. http://www.sophos.com/.

[25] SpamAssassin. The apache spamassassin project. http://spamassassin.apache.org/.

[26] The Editors. Product of the year: Spam? *Information Week*, January 2004.

[27] R. Twining, M. Williamson, M. Mowbray, and M. Rahmouni. Email prioritization: reducing delays on legitimate mail caused junk mail. In *USENIX Conference*, June 27–July 2 2004.

[28] M. Wong. What email forwarding services need to know about SPF. http://spf.pobox.com/emailforwarders.pdf.