Most of these mechanisms assume that incoming spam is detectable because it is sent at a high rate, and that slowing it down will reduce the amount of spam received. While it is possible for spam to arrive at a high rate, e.g. during a dictionary attack [1, 14] where a spammer bombards a server with messages addressed to random email addresses, our data will show that this generally is not the case. Most spam that is received by our servers is sent by servers from which only small numbers of messages are received, and those messages are not received at particularly high rates. This shows that rate-limiting mechanisms are unlikely to be effective.

The last pre-acceptance response is tempfailing. This is an SMTP control code that means "my server is temporarily unavailable, please try again later" to the sending server [21]. Well-configured mail servers will retry the mail later, but currently most spamming software and email-based viruses do not attempt retries. This is the premise of Greylisting [12], in which a server keeps a list of triples consisting of the sending host's IP address, sender's email address and recipient's email address, and tempfails mail that would generate new triples for some period of time. Since spamming software does not retry, this vastly reduces the amount of spam accepted by a server running Greylisting, while normal mail is subject to some delay. The weakness of this approach is that if (and when) spammers implement retries, this technique will become less effective.

Tempfailing is a useful strategy when used in combination with blacklists. It forces the spammer to maintain the IP address for longer, and reduces the rate that the spammer can send mails. This reduces the number of mails that are sent before the address is added to a blacklist.

Although tempfailing is a useful tool in the fight against spam, our data shows that the Greylisting approach is rather inefficient, delaying a large proportion of good mail, and requiring rather large numbers of triples. We suggest that evidence of spamming can be better obtained by looking at the history of mail sent by a particular server, and that this combined with tempfailing can be effective.

The main post-acceptance response is to look at the content of the mail message, scan it for viruses and examine the text for evidence of spam. Spam filtering can be accomplished by a variety of means, including collections of simple rules [25], Bayesian reasoning [22], and collaborative filtering [6]. It can also be implemented on the client, but is increasingly implemented on the server, often as an add-on to existing scanning for viruses.

None of these filtering mechanisms are infallible, and all suffer from the problem of false positives or incorrectly classified mail. In addition, the filters must be continually updated, as spammers develop new means to evade them [11, 20]. Moreover scanning mechanisms cause extra loading at the server, resulting in delays when the server is heavily loaded.

In this paper we introduce a new post-acceptance response, that of prioritizing the flow of good mail through the mail server, and processing mail suspected of being junk (virus, spam, or undeliverable) at reduced priority. The aim is that in times of heavy system load there will be prompt delivery of good mail, at the expense of delays to other mail.

3 Characteristics of email traffic

To evaluate existing responses and develop new ones, we collected data from the logfiles of two Internet-facing email servers in a large corporation, as detailed in Table 1. Server1 and server2 are the primary and secondary server for a single email domain. These servers ran a virus checker (Sophos [26]), and a spam filter (Spam Assassin [25]), using MailScanner [33], so for each SMTP transaction it is possible to discover from the log files whether received mail contained viruses or was flagged as spam. Server1 and server2 did not store mail for reading or check the addresses of recipients, but forwarded mail to other servers. Undeliverable mail was thus indicated by a failure to deliver to these servers, and was recorded in the logs. The logs also included records from incoming mail blocked because of real-time blacklists and other heuristics. This data gives a good picture of a server's-eye view of spam.

These servers are listed in public MX lookups, but also receive some mail from other mail servers within the corporation (they are part of an internal mail routing chain). Because the intent of this work was to combat spam entering a corporation, we removed the data corresponding to this "indirect" mail, leaving only the "direct" messages—interactions with other mail servers over the Internet.

Both servers subscribed to real-time blacklists, and performed other checks on the sender's address before accepting messages. These mechanisms rejected around 34–36% of incoming messages (see Table 2). However this is only partially effective at reducing spam, as

Table 1: Details of the data collected.

server	length (days)	number of messages	number of recipients
server1	69	855,228	1,229,459
server2	69	755,565	1,097,169

Table 2: Effectiveness of real-time blacklists and other blocking responses. While blacklists and other checks block around 34–36% of incoming messages, the accepted mail is still mostly junk, as shown in Table 3.

type	server1		server2	
	number	%	number	%
rbl	256,700	19	207,144	18
open relay	119,161	9	95,536	8
other checks	112,555	8	81,170	7
total blocked	488,416	36	383,850	34
total accepted	855,228	64	755,565	66
total attempts	1,344,960	100	1,139,415	100

Table 3: Breakdown of mail messages by type for each server. The percentages for spam, undeliverable and virus do not sum to 100%, as messages can fall into multiple types. The last two rows show totals of accepted good mail and accepted junk mail, where junk mail is any one of virus, spam or undeliverable.

type	server1		server2	
	number	%	number	%
good	260,348	30	262,941	35
spam	497,554	58	414,234	55
virus	2,749	0.3	2,371	0.3
undeliverable	364,487	43	298,169	39
total accepted good	260,348	30	262,941	35
total accepted junk	594,880	70	492,624	65

around 65–70% of the accepted mail is junk, as shown in Table 3.

Table 3 shows a breakdown of accepted messages. The values for spam and virus reflect messages that were flagged as such in the logs. A message was deemed undeliverable if on the first delivery attempt more than half of the recipients failed. The good messages are those that are left: not virus, spam or undeliverable. The table shows the enormous volume of mail that is junk (65–70% of all accepted mail). This is consistent with other measures of the prevalence of spam [15]. What is also surprising is the number of undeliverable messages. These appear to be mostly spam—of the 364,487 undeliverable messages for server1, only 94,735 (26%) were not classed as spam or virus. A likely cause is "trolling" [12, 1], where spammers send messages to automatically-generated usernames at known domains, hoping that these correspond to genuine email addresses. This figure is far too high to be explained by mistyped addresses, corrupted mailing lists or servers that are offline. Assuming that nearly all of the undeliverable messages are spam suggests that a significant proportion are not being classified correctly by the spam filter. This in turn suggests that the measure for "good" in the table is optimistic: it includes some spam messages. Overall these figures are alarming, as they show how many resources are wasted in passing junk emails through the email system.

Table 4 shows the number of servers that send (only) good mail, (only) junk mail, and a mixture, together with the number of messages sent. A small proportion of servers (11%) send only good mail, and these are responsible for a similar proportion of the messages (11%). By contrast, the overwhelming majority of servers send only junk mail (79%) but generate a relatively small proportion of the messages (48%), implying that each server sends few messages. The mixed class contains data for servers that sent at least one good and one junk mail. This includes those that send mostly good with the occasional junk (e.g. a mistyped address, a false positive spam detection), or those that send mostly junk with the occasional good (e.g. a spammer whose spam occasionally evades spam detection), or a more even mixture that might be obtained from an aggregating mail server (e.g. an ISP). This class is a small proportion of the servers, but generates a large proportion of the messages.

Figure 1 shows the same effect, but broken down by how many messages each server has sent. The figure was constructed by counting how many messages of each type (good/junk) each server sent, and plotting the cumulative percentage of total messages against the num-