Personalised, Collaborative Spam Filtering*

Alan Gray and Mads Haahr

Distributed Systems Group, Department of Computer Science, Trinity College Dublin, Ireland.

Abstract. The state of the art sees content-based filters tending towards collaborative filters, whereby email is filtered at the MTA with users feeding information back about false positives and negatives. While this improves the ability of the filter to track concept drift in spam over time, such approaches make assumptions implicit in centralised spam filtering, such as that all users consider the same email to be spam. In this paper, we detail and analyse these assumptions and describe how they affect spam filtering. We present an architecture for personalised, collaborative spam filtering and describe the design and implementation of proof-of-concept, peer-to-peer, signature-based system based on the architecture. The evaluation is based on real-world users employing the system as their spam-filtering tool. Preliminary analysis of the results indicates that the implementation is accurate and efficient.

1 Introduction

In response to the growing volume and variety of spam, spam filters have moved away from monolithic repositories situated on central servers towards dynamic knowledge bases located on local servers. The state of the art in spam filtering sees content-based filters tending towards collaborative filters, whereby email is filtered at the mail server using content-based techniques, with users feeding information back about false positives and false negatives. This feedback enables the spam filter to track concept drift in spam and to be retrained in the case of false positives.

While these filters can achieve statistically impressive accuracy rates, they remain prone to false positives, i.e., the erroneous classification of a legitimate email as spam. Underlying the current generation of spam filters are three implicit assumptions that, in our opinion, directly contribute to the presence of false positives. To address these, we propose the concept of personalised, collaborative spam filtering.

This paper is organised as follows: Section 2 discusses current trends in spam filtering. Implicit in all of these approaches are three assumptions, which are identified. Systems that advocate significant changes to the mail transport layer are considered beyond the scope of this paper and are omitted. Section 3 explains how personalised, collaborative spam filtering follows these trends and presents our architecture for personalised, collaborative spam filtering. A P2P proof-of-concept implementation is described in Section 4 and evaluated in Section 5. Section 6 briefly discusses related work. Section 7 presents our conclusions and outlines avenues for future work.

2 Background

Content-based filters are founded on the premise that it is possible to create a set of rules, exemplars or features that represent the "spamminess" of an email, and that if this is over some threshold, is considered to be spam. Such filters have been the focus of considerable interest, with work on rule-based filters [11], nearest-neighbour classifiers [16, 6], decision trees [3] and Bayesian classifiers [15, 1]. Initial implementations of these filters were centralised, but with spam comprising 50% of all email traffic [12, 14] they were replicated at the MTA level for performance reasons. As the knowledge base is now in the hands of the system administrators, it can be customised to suit the characteristic email and spam that individual domains receive. Users can feed information back about false positives and false negatives that enables the filter to be retrained. SpamAssassin [11] is perhaps the most prominent example of this approach. Thus the monolithic content-based filters have been developed towards a higher degree of collaboration as they have become decentralised.

* This research was supported by funding from Enterprise Ireland under grant no. CFTD/03/219.

Similarly, collaborative filters have been developed in the direction of content-based filters. In a collaborative system, when a user classifies an email as spam, a signature is computed on the email and added to the collective knowledge base. A signature is computed on every new email received and compared to the database of known spam. If the signature matches one in the database, it is deemed spam. The algorithm used to compute signatures here is key. If it is robust (i.e., ignores small randomisations in the text), then it is more suitable. In order for algorithms to be more robust (or "fuzzy"), they have been developed to be more content-aware so that unimportant discrepancies between emails do not change the signature. There are a number of very successful collaborative filters today: Vipul's Razor [13], Distributed Checksum Clearinghouse [18] and SpamNet [4].

Characteristics-based filters operate on the assumption that some facets of an email are highly indicative of spam. This is often done collaboratively. Characteristics on which emails are filtered include minimum number of recipients [18], whether the originating domain is an open or compromised server [19, 2], a known "spam-haven" [20], or a combination of several such factors [19, 9].

In our opinion, the assumptions listed below lead to reduced accuracy in centralised filters because emails are filtered inconsistently across different users, according to whether they are of interest to the user. These personalised views of what constitutes spam means that a centralised filter will cause false positives for users whose opinions differ from the majority. The assumptions that are implicitly made by centralised filters are:

- **All users classify email to be spam according to the accepted definition of UCE.** Most definitions assume UCE (Unsolicited Commercial Email) and spam to be synonymous. However, people do not classify emails as spam objectively purely on whether they adhere to a definition, but rather subjectively on whether the email is of interest to them.
- **All users consider the same emails to be spam.** Studies [8] show that people have their personal views on what constitutes spam. It is noteworthy that some consider email to be spam even if they have explicitly given the sender permission to contact them. This reflects some of the conundrums of legislative debates on spam.
- **All users are equally likely to receive the same spam.** A given email address will not appear on all spam lists and will therefore not receive all spam. By assuming that all spam messages are relevant to all users, each email must be checked against every known spam for all users, and not just the subset that a given user might receive. This can lead to suboptimal performance of the system.

3 The CASSANDRA Architecture

The Collaborative Anti-Spam System Allowing Node-Decentralised Research Algorithms (CASSANDRA) architecture allows the construction of personalised, collaborative spam filters. In this section we introduce the concept of personalised, collaborative spam filtering and present the CASSANDRA architecture.

3.1 Personalised, Collaborative Spam Filtering

Users receive spam because they are on mailing lists to which it is sent. Therefore, users need only be notified of spam that is sent to the lists they are on. A personalised, collaborative filter is one that delivers the most relevant spam notices to each user from the collection of all spam that is reported by members of the network. Additionally, only the emails that the user considers to be spam should be marked as such.

In order for a personalised, collaborative filter to work, every time a new spam is classified, a signature must be computed and propagated to those users most likely to receive a similar mail and to also consider it spam. This requires that information is compiled and maintained for each user as to the groups and other users it is similar to. The P2P architecture lends itself nicely to such a system, as there are unused resources situated at the fringes of the network that can be used for spam filtering purposes. These resources — specifically storage space, CPU cycles and highly accurate spam classifiers, in the form of users reading the email — enable larger and more complex spam solutions to be built. This follows the pattern of decentralisation and personalisation of spam filters. P2P systems can also provide better scalability, resilience and availability than centralised systems, and these features can be leveraged by a personalised, collaborative spam filter.

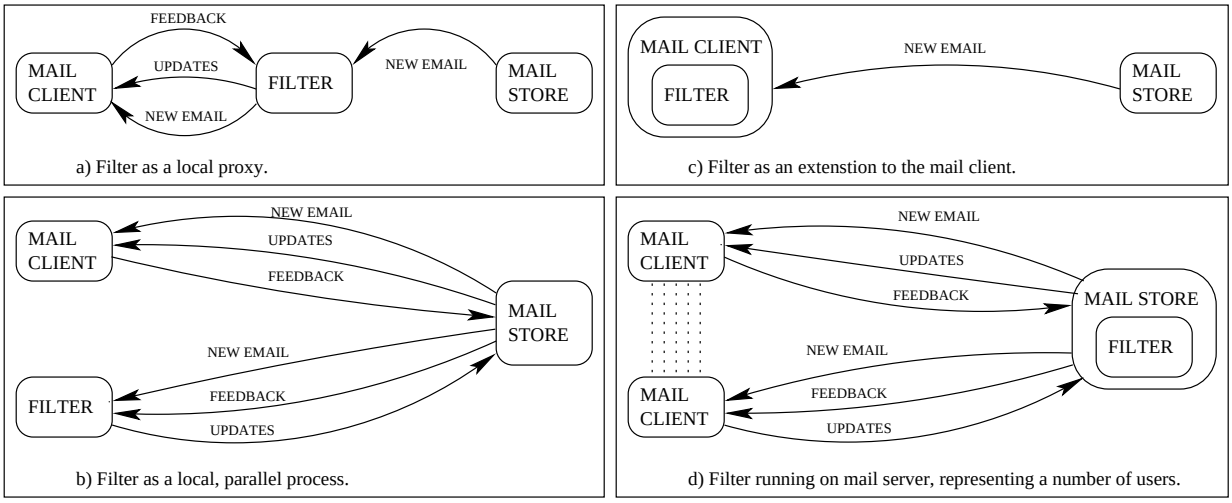

Fig. 1. Possible placements of filter.

Personalised, collaborative spam filtering has the benefit of being able to track concept drift in spam whilst minimising the working set of spam stored by any one user. Similar users will cluster together on the adaptive P2P network, reducing network traffic as relevant signatures are stored locally or nearby. The peers can be implemented as one filter representing a single user or as one filter representing the aggregate of a number of users. Some possibilities are shown in Fig. 1. A filter is comprised of several services to manage a peer's view of the network: Classifier Management Service (CMS); Algorithm Management Service (AMS); Peer Management Service (PMS); Trust Management Service (TMS); and Group Management Service (GMS).

3.2 Topology and Decision Based Routing

Topologically, CASSANDRA is an adaptive P2P overlay network. Adaptive P2P networks allow for the topology (i.e., the graph of connections between nodes) to change in response to the actions of the other peers on the network. This is due to DBR (Decision Based Routing) being used in unstructured P2P overlay networks, where any peer can decide to create or remove a connection between itself and any other peer. Adaptivity yields three useful features. Firstly, an adaptive network allows a peer to locally decide which other peers to whom it maintains connections. Hence, peers that receive similar emails and classify them the same way can gravitate towards each other and cluster together. Secondly, DBR allows the peers in the network to be heterogeneous in terms of decision making criteria, implementation and resources. Thirdly, adaptive P2P topologies are resilient to failure and attack because it is extremely difficult for an attacker to get a complete picture of the system at any given time.

3.3 Algorithms and Classifiers

Classifiers are used to identify email as spam. Examples of classifiers in a signature-based filter would be signatures, whereas in a rule-based filter would be the rules themselves. As the filter is used, the CMS maintains information about the most useful classifiers. A peer shares its best classifiers with other peers to whom it has clustered. New classifiers are created in response to false positives/negatives and added to the knowledge-base of a peer. The next time a peer shares classifiers, any new ones are sent to neighbouring peers. Thus new classifiers are propagated through the network.

Classifiers are associated with the algorithms that created them. The only constraint on the classifiers (and hence on the type of collaborative filter that can be implemented) is that they must be expressible in XML. The AMS tracks and handles the algorithms that the peer uses. It is responsible for creating new classifiers and ensuring that signatures are compared only to those generated by the same algorithm. In the

event that a peer does not know an algorithm with which a classifier was computed, the AMS can locate and obtain the module for it.

3.4 Clustering and Collaboration

The PMS manages this peer's view of the network. It maintains information about the peers in the network to whom it is connected as well as those with whom it has had interactions in the past. The exact semantics of what it means to cluster with other peers are not prescribed by the architecture, but left to individual implementations. For instance, two peers might consider the amount of spam they have in common or network locality to be dominant features when clustering. Malicious peers will get pushed out of clusters, as the surrounding peers will detect that the peer is not acting similarly to them, and tend not to cluster with them. Also noteworthy is the fact that connections between peers are unidirectional. This is due to DBR, as two peers' opinions of their shared relationship can be different (for example, a lazy peer would want to cluster with active peers, but not vice versa). The TMS manages information pertaining to past interactions with other peers, i.e., how much they are trusted to share good classifiers. The GMS manages this peer's view of which other peers are likely to be on the same spammers' lists as it is.

3.5 Inter-Peer Communication

The CASSANDRA architecture does not prescribe the transport layer or the algorithms used when sending messages. However, XML schemas are defined for the messages in order to ensure interoperability of peers. There are three occasions upon which peers communicate with each other:

- **Joining:** In order to join the network, a node sends a Ping to a peer that is already in the network. The extant peer replies with a Share message, which is a list of the best peers and classifiers from its perspective. The extant peer then forwards the Ping to other peers, according to the algorithm used in the system.
- **Sharing:** A Share message, containing information about classifiers and peers, is sent by a peer when it determines it necessary. This can be after a fixed interval, at application start-up or shut-down, after a new classifier has been created, or any combination of these. The recipients of a Share are those peers whom the sending peer deems most likely to benefit from the new information.
- **Revoking:** Any peer can revoke any classifier at any time and sends a Revoke message when it does so. This will typically be because the classifier has resulted in a false positive for the user. The recipients of a revoke message can also choose to revoke the classifier or ignore it, based on past interactions with the revoking peer.

4 Proof-of-Concept Implementation

The filter was implemented as a single-user filter that runs on a user's computer in parallel to their normal MUA, as per (b) in Fig. 1 above. It manipulates the user's mail-store over IMAP using the JavaMail API. The user does not interact directly with the filter. When the user is configuring the filter, he/she specifies one folder on his mail store to be a spam folder. The user can identify false negatives simply by moving emails into the spam folder and revoke false positives by moving them out of the spam folder, by using their MUA as normal.

4.1 Filter Management

A custom header is added to emails to signify that the filter has classified them as spam or nonspam. The filter operates by periodically checking the mail store to see if the number of emails in any of the folders has changed. If so, the filter manager is run. Any emails that have no custom headers have signatures computed on them, which are compared to the list of known spam. If there is a match, then the email is given a spam header and moved to the spam folder. Otherwise it is given a nonspam header and delivered normally. All emails that are in the spam folder and have nonspam headers are false negatives, and any emails that are not in the spam folder and have spam headers are false positives, as shown in Fig. 2 below.

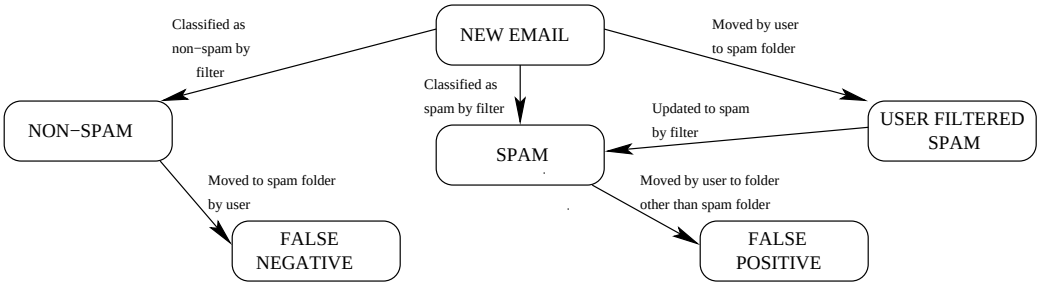

Fig. 2. Email state diagram.

4.2 Algorithms and Classifiers

This system is signature-based and uses only one algorithm, called Approximate Text Addressing (ATA). ATA generates a block text fingerprint on emails and has been shown to be a robust and accurate algorithm [21]. A hit is when a signature computed on a new email matches a known signature. Each peer keeps a list of the 1000 most recent signatures upon which it has had a hit, ordered by the time and date it last hit. The email is tagged as spam, and the last hit time updated. Signatures are replaced using LRU replacement when more than 1000 signatures are known.

4.3 Clustering and Collaboration

Each peer maintains a list of peers with whom it has interacted. For each peer, it maintains a value in the range [0.0, 1.0] for both similarity and trust. Similarity is a measure of the UCE in common and trust a measure of how likely the peer is to perform the same classification of the common email. These values are modified on hits, misses and revokes. A miss is when a signature is replaced without having had a hit and a revoke is when a signature is revoked by the peer or one of its neighbours. Peers are ranked by the product of these values.

Upon hits, the value for similarity and trust are both increased by 10% of the amount it is less than 1.0 for peers that have recommended this signature to us. This enables peers to quickly build up a good reputation for themselves. On misses, the value for similarity for the recommending peers is reduced by 10% of its current value, as the signature was not relevant to the peer. On revokes, recommending peers lose 25% of their current value for trust. This is quite drastic, but it ensures that peers that get the same UCE but tag it differently do not cluster, as these are the peers most likely to cause false positives for each other.

4.4 Inter-Peer Communication

Peers communicate via XML messages over SMTP. SMTP was chosen because peers are easily identified by their email addresses and SMTP enables asynchronous communication without loss of messages if a peer is offline. It is also natural to assume that SMTP will be available, whereas other transport layers, such as TCP may not be. To the authors' knowledge, this is the only spam filter that utilises SMTP in this fashion. Peers communicate according to the following rules:

- **Joining:** A node sends a Ping to a peer in the network, which relays it on to five peers it knows, with a time-to-live of 3 hops. Every peer who receives the Ping sends a Share to the newcomer.
- **Sharing:** Every time a false negative is identified, a new signature is computed and added to the list. A new share object, comprised of the most recent 10 signatures and top 5 peers, is sent to the top 5 peers (i.e. the closest peers in the network).
- **Revoking:** Every time a false positive is identified, a Revoke is sent out with the offending signature to the top 5 peers. Upon receiving a revoke, if the revoking peer is ranked higher than any peer that has had a hit with that signature (including the receiving peer), then the recommendation is accepted and the signature revoked.

5 Preliminary Results and Evaluation

Here we present preliminary results originating from a two week long case study of six people using the implementation presented in Section 4 as their spam filter. The filter was used on a range of MUAs and operating systems: Outlook 2003 on Windows XP Professional; Evolution 1.4 and Mozilla Thunderbird 0.5 on Fedora 1.0; and Mozilla 1.6 on Debian "Sid" (unstable).

Peer ID	Peer 1		Peer 2		Peer 3		Peer 4		Peer 5	
	Sim	Tru	Sim	Tru	Sim	Tru	Sim	Tru	Sim	Tru
1	1.0	1.0	0.3439	0.3439	0.0	0.0	0.0	0.0	0.1	0.1
2	0.9938	0.9721	1.0	1.0	0.1	0.075	0.271	0.271	0.0	0.0
3	0.9953	0.9728	0.7175	0.5579	1.0	1.0	0.3439	0.3439	0.1	0.075
4	0.9953	0.9728	0.7456	0.5738	0.6513	0.489	1.0	1.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 1. Peers' values for similarity and trust.

Peer ID	Peers	Accepted Signatures	Revoked Signatures
1	6	13	2
2	6	13	2
3	6	12	3
4	6	13	1
5	6	11	1

Table 2. Peer's knowledge of signatures and other peers in the network.

Peer ID	Non-Protocol Emails Received			Protocol Emails Received			
	Spam	NonSpam	Total	Pings	Shares	Revokes	Total
1	58	227	285	13	25	10	48
2	18	121	139	8	17	8	33
3	19	711	730	9	10	7	26
4	10	1959	1969	2	24	6	32
5	2	94	96	0	10	1	13

Table 3. Protocol and non-protocol email received by each peer.

5.1 Clustering and Collaboration

Table 1 shows the values for similarity (Sim) and trust (Tru) that the peers have for each other. It is read across to see how a node is ranked by its peers and down to see how a node ranks its peers. The table shows that some peers are ranked highly by many peers and also that some peers have information about many peers. This indicates that the clustering algorithm works. Early reporters (peers that report new spam to the group) are the only ones who generate share messages. Therefore, information about these peers is propagated through the network. "Freeloading" peers (those that have not contributed to the system whilst benefiting from it) are ranked lowly, because they have not shared signatures. As the system grows, these peers will be pushed to the edges of clusters. It can be seen from Table 2 that the freeloading peers know similar numbers of peers and signatures to the other peers. Thus the information about the signatures flows from regions of high to low activity in the network, showing the collaborative nature of the system. This may lead to inactive peers being isolated in the network, as other peers do not have information about them. To increase the flow of information from low activity regions, the sharing protocol can be extended to include a reply Share message being sent when a peer receives a Share.

5.2 Accuracy and Scalability

Table 4 summarises the false positive and negative rate for each user. Accuracy is defined to be the percentage of a user's mail that is filtered correctly, i.e. as the user him/herself would filter it. As can be expected, the false negative rate is highest for those peers that generated the most signatures. The figures in Table 4 describe moderately successful rates for false positives, negatives and overall accuracy.

Peer ID	False Negatives	False Positives	Total Email	% False Positives	% False Negatives	Accuracy
1	16	3	285	1.05%	5.61%	93.33%
2	0	2	139	1.44%	0.00%	98.56%
3	1	3	730	0.41%	0.14%	99.45%
4	1	7	1969	0.36%	0.05%	99.59%
5	0	1	96	1.04%	0.00%	98.96%

Table 4. Accuracy of the filter for each peer.

A system is described as scalable if it will remain effective when there is a significant increase in the number of resources and users [5]. In the context of our prototype, the number of protocol messages must remain low, while the accuracy for each user must not decrease. Table 3 shows that peers receive approximately similar numbers of protocol messages. Share messages are sent only when new unseen spam is received by a peer. Therefore, as the knowledge bases at the peers are filled, less and less unseen spam will be received. When there are no new mutations of spam being received, the protocol message overhead will be nil, save for peers joining and leaving the network. Collaborative filters suffer from what is known as the "pump-priming" problem. In other words, the effectiveness is low for small knowledge-base and number of users, but increases dramatically as these values increase. Therefore, we expect (but have yet to test) that the effectiveness of the proof-of-concept system will increase with scale.

5.3 Network Resilience and Security Analysis

The network was formed by nodes joining via randomly selected peers in the network. Table 1 shows that all peers knew of the existence of each other. This means that from a fragmented beginning, the network became more fully connected over time as the peers shared information about each other, displaying the implementation's resilience. The network is intended to have an adaptive topology as peers cluster towards each other, yielding this self-healing characteristic.

SMTP is unauthenticated and hence the prototype system is susceptible to sender-spoofing, whereby a malicious peer sends fake recommendations, purporting to be from another peer. Cryptographically strong authentication can be achieved by using a decentralised authentication tool such as the Claim Tool Kit [17]. Such a system does sender entity recognition by the exchange of hashes of previous emails, ensuring that sender-spoofing cannot work, unless a spammer has compromised the channel in a so-called "man-in-the-middle" attack. Where there is a man-in-the-middle attack, it is localised to the channel that is compromised (i.e. between two users, or on a mail-server). If we assume that hacking into a user's machine/mail server is difficult and resource consuming, it is unprofitable for an attacker to do widescale sender-spoofing.

The prototype is also susceptible to the Sybil attack [7], where one entity (a spammer, say) manifests itself as many peers who collaborate to undermine the system. Such an attack is viable where the cost of creating and using these identities is less than the benefit gained. The Claim Tool Kit protects against widescale Sybil attack, because computing hashes on previous shared emails is computationally expensive, as is creating signatures on the spam email that is to be put into the previously shared recommendations.

6 Related Work

SpamNet [4] claims to be a P2P spam filter, but is probably more accurately described as a centralised, collaborative filter in the Napster architecture (signatures are stored centrally). To the authors' knowledge, the only other example of a decentralised spam filter is that presented in [21]. The system uses Approximate Text Addressing, a variant of the block-text fingerprinting first introduced in [10], to compute signatures on reported spam. The system is designed to overlay structured P2P topologies, where each peer manages a subset of the key-space. Each signature corresponds to a location in the key-space. When an email is received, a signature is computed and the peer who manages the key-space that it maps to is queried to determine whether the email is spam or not. This works well for a LAN-sized network, but has the drawback that every

user generates network traffic for every email they receive. It is possible that the network, or parts of it, can become congested if too many emails are received at once, or if many peers receive the same mass-mail (spam or not) and concurrently poll the node to which the signature routes. This is true for any structured P2P system, as they all use Key Based Routing (KBR).

7 Conclusions and Future Work

We have presented the concept of personalised, collaborative spam filtering. The CASSANDRA architecture for personalised, collaborative spam filtering was detailed, along with a proof-of-concept, peer-to-peer, signature-based implementation. The case study presented above shows that for a small system starting from an empty knowledge base using quite simplistic clustering and trust processes, moderately successful false positive and negative rates can be achieved. We are currently working on deploying the spam filter in a long running case study with a larger set of users and expect improved performance and effectiveness, due to the economies of scale and once the "pump-priming" problem (as discussed above) is overcome. Other interesting avenues for future work include implementing different filters, such as collaborative rule-based or case-based filters and the development of more sophisticated clustering methods.

References

1. Ion Androutsopoulos, John Koutsias, Konstantinos V. Chandrinos, Georgios Paliouras, and Constantine D. Spyropoulos. An Evaluation of Naive Bayesian Anti-Spam Filtering. In *Proceedings of the workshop on Machine Learning in the New Information Age*, 2000.
2. Blitzed. The Open Proxy Monitor. http://opm.blitzed.org/info, 2004.
3. Xavier Carreras and Lluís 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. Cloudmark. SpamNet. http://www.cloudmark.com, 2004.
5. George Coulouris, Jean Dollimore, and Tim Kindberg. *Distributed Systems: Concepts and Design (3rd Edition)*. Addison-Wesley, 2001.
6. Pádraig Cunningham, Niamh Nowlan, Sarah Jane Delany, and Mads Haahr. A Cased-Based Approach to Spam Filtering that can Track Concept Drift. Technical Report TCD-CS-2003-16, Trinity College Dublin, Ireland, 2003.
7. John R. Douceur. The sybil attack. In *The IPTPS02 Workshop*, Cambridge, MA (USA), mar 2002.
8. Deborah Fallows. Spam: How it is hurting email and degrading life on the Internet. *Pew Internet and American Life Project*, October 2003.
9. Mail Abuse Prevention System LLC. MAPS RBL - Mail Abuse Prevention System Realtime Blackhole List. http://mail.abuse.net/rbl+/, 2004.
10. Udi Manber. Finding Similar Files in a Large File System. In *Proceedings of the USENIX Winter 1994 Technical Conference*, pages 1–10, San Fransisco, CA, USA, 17–21 1994.
11. Justin Mason. The SpamAssassin Homepage. http://spamassassin.org/index.html, 2004.
12. MessageLabs. Intelligence Monthly Report, July 2003.
13. Vipul Ved Prakash. Vipul's Razor. http://razor.sourceforge.net, 2004.
14. Sara Radicati and Masha Khmartseva. The IT cost of spam. *The Messaging Technology Report*, 12, August 2003.
15. Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. A Bayesian Approach to Filtering Junk E-mail. AAAI Tech. Report WS-98-05. In *Workshop on Learning for Text Categorization*, Madison, WI, 1998.
16. Georgios Sakkis, Ion Androutsopoulos, Georgios Paliouras, Vangelis Karkaletsis, Constantine Spyropoulos, and Panagiotis Stamatopoulos. A Memory-Based Approach to Anti-Spam Filtering for Mailing Lists. *Information Retrieval*, 6:49–73, 2003.
17. Jean-Marc Seigneur and Christian Damsgaard Jensen. The Claim Tool Kit for Ad-hoc Recognition of Peer Entities. *The Journal of Science of Computer Programming*, 2004.
18. Rhyolite Software. Distributed Checksum Clearinghouse. http://www.rhyolite.com/anti-spam/dcc/, 2004.
19. The Spamhaus Project. The Exploits Block List. http://www.spamhaus.org/xbl/index.lasso, 2004.
20. The Spamhaus Project. The Spamhaus Block List. http://www.spamhaus.org/sbl/, 2004.
21. Feng Zhou, Li Zhuang, Ben Y. Zhao, Ling Huang, Anthony D. Joseph, and John D. Kubiatowicz. Approximate object location and spam filtering on peer-to-peer systems. In *ACM/IFIP/USENIX International Middleware Conference*, June 2003.