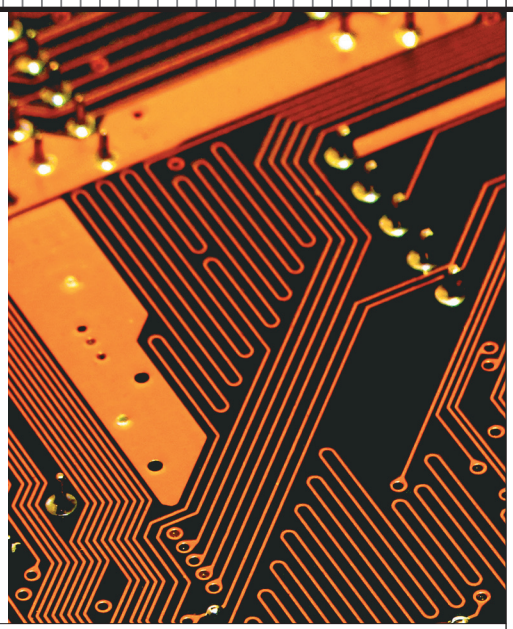
Channel E-mail: A Sociotechnical Response to Spam



Brian Whitworth and Tong Liu
Massey University—Albany, New Zealand

Spam wastes Internet processing, bandwidth, and storage. Like many other sociotechnical problems computing today faces, it's not solvable by purely technical approaches like filters or social responses like passing laws. Channel e-mail offers a solution that can enable social as well as technical communication efficiency.

Today about 80 percent of the more than 460 billion e-mails sent per year are spam, electronic garbage that wastes Internet bandwidth, storage, and processing.[1] Even spam "caught" by filters has already been transmitted, downloaded, processed, and stored, using up resources on users' computers and the Internet. In May 2003, spam exceeded nonspam for the first time[2]—an Internet service provider (ISP) using one server for e-mail customers needed another just for spam mostly deleted on arrival, a cost passed on to customers.

The growth of spam traffic has increased dramatically in recent years: from 20 to 40 percent in 2002-2003, 60-70 percent in 2004, 86 percent in 2006, and 92 percent in 2007.[3-5] The problem has expanded in complexity as well, as image spam now bypasses text filters, spambots harvest website e-mails, and real users become "zombie" spammers. Spam has now migrated to other applications, such as instant messaging and chat (SPIM and SPAT, respectively), and to other platforms like cell phones. Spam now surpasses viruses as the number one unwanted network intrusion, and it has always been the biggest e-mail complaint.

The economic impact of spam is staggering. In 2003, spam cost US companies $10 billion in lost productivity; a 2004 study calculated that the average cost of spam per year per employee to US organizations was $1,934, ignoring IT staff, hardware, software, and bandwidth costs; and a 2005 estimate put spam costs at about $50 billion globally and rising.[6,7]

While the global e-mail system is technically efficient—it transmits information well—it's socially inefficient—it transmits meaning increasingly poorly. It's hardly surprising that a system designed to purely technical requirements before the Internet became a social medium now doesn't perform well socially. Nonetheless, that e-mail technology now *enables* the antisocial communication we call spam is an issue the online community must deal with at some point.

Spam is an old social problem in new technical clothes, essentially an electronic "tragedy of the commons."[8] It's a *sociotechnical* problem, so neither technological responses like filters nor social responses like laws can resolve it. A 2004 article predicted that while filters may limit inbox spam, transmitted spam will inevitably rise to over 95 percent within a decade.[8] Unfortunately, this prediction seems to be coming true all too soon.

Sociotechnical problems like spam require sociotechnical solutions. We propose channel e-mail as an example of such a solution.

0018-9162/09/$25.00 © 2009 IEEE Published by the IEEE Computer Society

TECHNOLOGY-BASED RESPONSES

Technology-based responses oppose spam using code without considering social issues.

Filters

Spam filters, the common technology response to spam, aim to identify spam on arrival and place it in the trashcan for deletion. Over time they have adopted increasingly sophisticated methods like machine learning, compression techniques, and advanced similarity-matching methods.[9] However, as improved filters sent more spam to trashcans, spammers countered the reduction with more spam, and as filters became more intelligent, so did spammers. For example, as machine-learning filters identified spam words like "free," spammers wrote "f-r-e-e" or inserted blank HTML comments like "f<!---->ree," which became "free" when rendered.

The lists approach checks e-mails against white lists of nonspammers, black lists of spammers, or both.

Spammers can now bypass text detection entirely with image spam, sending images inside random innocuous e-mails impervious to text checks. Against providers with image-matching filters, spammers can randomize image content or, if providers block Web image e-mails, embed images in messages or break them into multiple pieces to be reassembled when the recipient renders the e-mail.[9]

In these "spam wars," the advantage shifts back and forth between filters and spammers, but the real outcome is that transmitted spam steadily grows and degrades our common communication system. There is no end to this arms race because spam is "almost impossible to define."[2] Filters may even exacerbate the problem by letting users sit behind filter walls unaware of the steadily rising tide of transmitted spam that consumes Internet resources whether they see it or not.

Pretransmission filtering could reduce spam, but all filters have false positives—real e-mail filtered as spam. A pretransmission e-mail filter wouldn't notify the sender, or spammers could tailor their spam to the filter; nor would the receiver know, as the message isn't sent. Users could not rescue real messages from their spam box, as they do now. If a valid e-mail that accidentally used "spam words" was prefiltered, neither sender nor receiver would find out. Receivers could ignore e-mails and claim "the filter took it," and users would lose confidence in e-mail without the postal system ethic that "the mail will get through."

Lists

The lists approach checks e-mails against white lists of nonspammers, black lists of spammers, or both. Black lists grow endlessly as spammers change identities or "spoof" real users (as zombie machines), using an account until it is blacklisted before moving on to another. The administrative effort to create and maintain lists means most individuals don't bother, but they're common at the ISP level. Yet if an ISP is blacklisted, innocent users have their messages blocked too. Also, this approach doesn't avoid *community spam*—unwanted messages from acquaintances or coworkers.

The logical extension of the lists approach is centralized coordination. The Tripoli (from "Triple-E"—Empowered E-Mail Environment)[10] method links a third-party-certified encrypted authentication token with every e-mail. However, that trusted third parties are institutional bodies raises Juvenal's question, "Who watches the watchers?" Will major e-mail stakeholders like the Direct Marketing Association, Microsoft, or Yahoo guarantee that e-mail isn't spam? If so, wouldn't they naturally exempt their own "useful services"? A centralized e-mail custodian could both let itself in and keep competitors out, a concentration of power that invites corruption.

Graylisting uses a combination of black and white lists to identify new e-mail addresses and automatically reject them temporarily on the grounds that spammers will move on while real e-mail senders will try again.[11] However, with this technique even temporary rejections for one to four hours can make messages disappear into an e-mail limbo, creating problems for, say, people waiting to receive passwords from websites.

Challenges

Challenge defenses essentially require new e-mail senders to prove they're people by answering questions supposedly easy for humans but not computers—for example, MailBlocks asks, "Are you really a person? If so, type the number in this graphic." However, computer pattern recognition AI can already meet such challenges.[9] Also, if everyone challenges, mutual challenges can form a challenge deadlock, and the graphic challenge transmissions use up extra resources. Such methods work, but some find continual challenges to their humanity while trying to converse annoying and rude.

Starting over

Because spammers network and trade addresses, older e-mails attract more spam and get on more spam lists. These trails can be broken by getting a new Hotmail or Gmail account, but unfortunately this breaks friendly connections too. Disconnecting negative social links also disconnects positive ones. Yet most people have multiple

e-mail addresses, and sending to an outdated e-mail is a common delivery problem.

SOCIALLY BASED RESPONSES

Socially based responses oppose spam with social methods like laws that punish spammers; they don't change e-mail code, although they might use the Internet to find culprits.

Spam the spammers

In primitive societies individuals mete out justice by eye-for-an-eye vendettas, which make antisocial acts less profitable as future losses from revenge attacks can cancel current gains. Axelrod's prisoner's dilemma computer tournament illustrated the principle, as the most successful strategy—tit-for-tat—paid back defaulters in kind.[12] Similarly, users responded to companies faxing annoying unsolicited messages by bombing them with return faxes, shutting down their fax machine. Stanford law professor Lawrence Lessig once suggested a bounty on spammers "like … in the Old West,"[13] but in an online society of vigilantes, a false-positive Internet rumor could shut down a good company. Also, because Internet spammers usually don't accept replies, spam counterattacks go nowhere.

Laws

Modern societies avoid the problems of vendettas by administering justice through the state, not individuals. Police, courts, and sanctions aim to make antisocial acts unprofitable, but antispam laws have been ineffectual for several reasons:[14]

- Physical laws may not transfer online—for example, what is online trespass?
- Virtual worlds change faster than laws can form—for example, new functions like cookies outstrip their assimilation into law.
- In cyberspace, programmers can bypass laws—for example, justice systems can't identify spoofed e-mail sources.
- Laws have limited jurisdiction—for example, US law applies only on US soil.

Because cyberspace transcends national borders, legislation like the US CAN-SPAM (Controlling the Assault of Non-Solicited Pornography and Marketing) Act and Australia's Spam Act, both passed in 2003, has failed to stop the e-mail inundation. Countries could "nationalize" their Internet to gain national control, but this would collapse the electronic community into national "tribes" and reduce global synergy. Legal prosecutions require physical evidence, an accused, and a plaintiff; however, spam begins and ends in cyberspace and is easily spoofed at source, and the plaintiff is everyone with an e-mail account. What penalties apply when each individual loses so little? The law is just too limited, slow, and impotent to deal with the global spam challenge.

SOCIOTECHNICAL RESPONSES

Social or technical responses alone seem powerless against sociotechnical problems like spam. Yet as physical architectures like doors and walls support social justice in physical society, so can technical architectures support communication fairness online. The sociotechnical approach involves first identifying a desired social principle, then translating it into information terms, and finally designing and implementing technology to these social requirements.

Stopping spammers by slowing the e-mail flow with unneeded charges or pointless calculations is akin to burning down your house to prevent break-ins.

Instituting an e-mail charge illustrates one sociotechnical response to spam. Its initiating socioeconomic principle is that people minimize costs, which suggests a system that hits spammers in their pockets. For example, every transmission could extract a micropayment or require senders to compute a time-costly function trivial for all but spammers, who would find the cost excessive.[15] This essentially increases transmission costs, but charging for e-mail contradicts the social principle that inspired the Internet in the first place: that fast, easy, and free communication creates social synergies for all.

An e-mail charge that reduces spam would also reduce legitimate e-mail. Stopping spammers by slowing the e-mail flow with unneeded charges or pointless calculations is akin to burning down your house to prevent break-ins. Communities should increase not decrease social synergy.

Also, how can one justify to people introducing a charge for services they already have? An Internet toll adds no new service, as e-mail already works without it. Making the Internet a field of profit also opens it to bureaucratic corruption. If senders pay receivers, and each e-mail transfers money, who administers the system and sets the charge rate? Is this charge effectively an e-mail tax? Who then will collect this tax and govern the world of online e-mail?

Spam works because e-mail costs so little, as we work as a community, but this is also why the Internet works, and its decentralization may be why it has largely resisted corruption—so far. The problem with an e-mail charge is not that it won't work, but that it's anti-Internet. A solution is needed that reduces spam but leaves the Internet advantage intact.

WHY SPAM WORKS

An effective sociotechnical response to spam must recognize the conditions that enable it:

- *Nonzero response percentage*. With a sucker borne online every minute, whatever the pitch, someone always buys.
- *Low-cost mass communication*. Sending 20, 20,000, or 20 million e-mails a day costs about the same.
- *No consequences*. Anonymous spammers make problems for others that don't affect them.
- *Communication unfairness*. E-mail lets senders place messages directly into receivers' inboxes.

If the response percentage is always positive, if sending more messages involves virtually no extra cost, if e-mail anonymity means no consequences, and if the technology allows unilateral "communication," isn't spam inevitable? The current email system virtually enables spam.

Unlike filters, lists, and challenges, channel e-mail doesn't target spammers but treats everyone the same.

To illustrate, if spammers typically send several hundred million e-mails a day, even with filters 99 percent successful (which they aren't), only 100 takers per 10 million requests—a 0.001 percent hit rate—is still profitable. Under these conditions, the logical spammer target is *all* Internet users. For the more than one billion global e-mail users, the predicted end point is a system that "efficiently" sends and receives trillions of messages, most of which are automatically deleted on arrival by computer filters. The filter barricade stopgap will become unsustainable as spam transmissions increase in number, size, and type.

Society denies antisocial acts by laws and punitive sanctions, but this strategy fails online where "code is law," as Lessig notes.[16] However, if code is law, and we control the code, why not use it as such? The danger, as Lessig also points out, is that software control could lead to an online police state, where freedom of expression is lost. The solution, argued elsewhere,[14] is for technology to encourage synergy rather than control antisocial acts—let code support politeness rather than deny spam.

CHANNEL E-MAIL

Unlike filters, lists, and challenges, channel e-mail doesn't target spammers but treats everyone the same. Its design enables polite conversation, which naturally combats spam.[17] The price, that we must all be considerate, seems worth paying.

Conversational requirements

On a technical level e-mail is asynchronous messaging, but on a human level it's a conversation, with all that implies. The conversational equivalent of spam is haranguing a stranger in the street to buy a product, or filibustering a legislative body. In contrast most people are polite, as even friends ask, "May I talk to you?" A human conversation is a mutual agreement, not a one-way transmission. Yet e-mail by design lets any sender put messages directly into any receiver's inbox—that is, it allows unilateral communication. The inadvertent design flaw, of giving all rights to senders and none to receivers, underlies the current spam problem.[8]

A sociotechnical design would share communication rights[18] by ensuring that:

- e-mail conversations require mutual consent;
- users have the right to be left alone and can refuse to converse;
- anyone can request to converse, given brief requests that identify sender and purpose;
- users converse in a turn-taking fashion, without further introductions; and
- any party can leave the conversation at any time.

Conversation protocol

Channel e-mail supports these requirements by creating a channel entity above any messages sent. Instead of managing messages users manage channels, each a conversation of many messages. A channel created by mutual consent grants parties the right to freely exchange messages, as in current e-mail. If no channel is open, users must negotiate one by channel request "pings"—small messages containing permissions. Opening a channel is a separate step from sending a message, like the handshaking before face-to-face conversations, or that of synchronous communication. This handshaking can be automated, so users just send messages while underneath the computer manages the permissions.

Instead of the current "send and forget" one-step protocol, channel e-mail is multistep:

1. *Channel request/permission*: a conversation request (A to B).
2. *Channel permission:* a permission reply (B to A).
3. *Message transmissions:* mutual conversational messages.
4. *Channel closure:* either party closes the channel.

Step 3 messages use the permissions of steps 1 and 2 to avoid further channel requests. Channels are defined by

participants, not message topic, content, or attachments. Channel control isn't just the right to tediously reject e-mails one by one, but the right to close a channel entirely, including any future messages from that source.

Aspects of this approach are already in practice. For example, Hotmail recognizes

- *safe senders*: senders granted a channel to send e-mail; and
- *blocked senders*: senders blocked from sending e-mail—that is, a closed channel.

Similarly, DiffMail handles spam using the following classifications:[18]

- *regular contacts:* message is sent (pushed) to receiver inbox;
- *known spammers:* message isn't delivered; and
- *unclassified:* message is retrieved (pulled) by receivers by choice.

Channel e-mail likewise classifies channels into

- *channel open:* always accept;
- *channel closed:* always reject; and
- *unclassified:* ask me each time.

The default would be "Accept all," as it's closest to the current state. While most list approaches are centralized, channel e-mail devolves communication control to users. As well as being protected by ISP black lists, channel e-mail gives users their own accept/reject lists.

List maintenance, a problem for centralized lists, occurs automatically by normal use in channel e-mail—any sent message opens a channel, and any rejected message closes the channel. Also, while ISPs need consensus to change their lists, individual users can open or close channels directly.

EVALUATION

We evaluated the channel e-mail design by theoretical calculation, simulation, and usability testing.

Theoretical social efficiency

While a spammed network may be technically efficient (in bytes/second), it's socially inefficient if most of its transmitted messages are spam. *Social efficiency* (SE) can be defined as the proportion of network resources sending socially useful bytes over a given period: nonspam bytes sent/total bytes sent. When SE = 100 percent, all network resources are used for nonspam messages; for, say, SE = 40 percent, only 40 percent of network capacity is used to transmit useful messages. A network can thus be technically efficient (transmits a lot of information quickly) but socially inefficient (mostly sends spam no one wants).

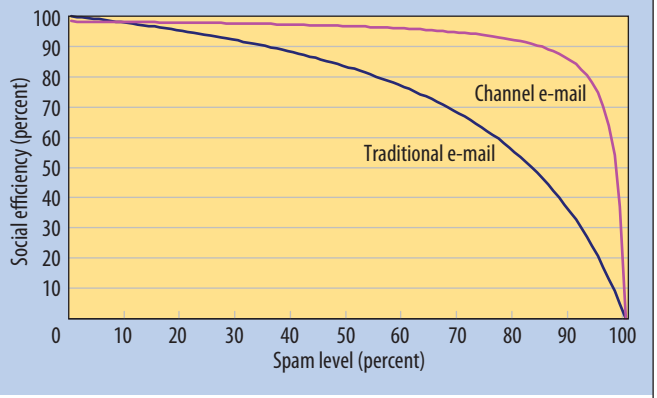


Figure 1. Social efficiency of traditional versus channel e-mail by spam rate. Traditional e-mail is initially more efficient, but as spam increases, its efficiency declines rapidly, while channel e-mail is more stable under spam load.

In a simple model with an average e-mail size of E bytes, an average spam size of S bytes, and spam X percent of all e-mail messages, the social efficiency of traditional one-step e-mail is useful message bytes transmitted divided by the total bytes sent:

$$SE_{TraditionalEmail} = E * (1 - X)/(E * (1 - X) + S * X)$$

Channel e-mail has the additional overhead of channel permissions. If the channel ping size is R bytes, and N percent of valid e-mails are new channel requests, channel e-mail's social efficiency is

$$SE_{ChannelEmail} = E * (1 - X)/(E *(1 - X) + 2 * R * N * (1 - X) + 2 * R * X),$$

where spam channel requests are by definition rejected.

Figure 1 compares traditional and channel e-mail social efficiency, for an average e-mail size E = 59 Kbytes, an average spam size S = 12 Kbytes, an estimated ping size R = 0.5 Kbyte, and new contacts N = 25 percent. Traditional e-mail is initially more efficient, but as spam increases, its efficiency declines rapidly, while channel e-mail is more stable under spam load.

Table 1 shows the percentage of network resources saved by channel e-mail compared to traditional e-mail. Once spam exceeds 10 percent, channel e-mail performs better, and at 80 percent spam, it saves over one-third of network resources, allowing an e-mail ISP to replace three servers with two.

Figure 2 shows traditional and channel e-mail social efficiency by spam size for an 80 percent spam load. While standard e-mail efficiency declines rapidly as spam size increases, channel e-mail is unaffected. This predicts that

Table 1. Network resources saved: traditional versus channel e-mail.

Spam (percent)	Social efficiency (percent)		Network resources saved (percent)
	Traditional e-mail	Channel e-mail	
0	100.0	99.6	-0.4
10	97.8	99.4	1.6
20	95.2	99.2	4.0
30	92.0	98.9	6.9
40	88.1	98.5	10.4
50	83.1	97.9	14.8
60	76.6	97.1	20.5
70	67.8	95.8	287.0
80	55.2	93.3	38.1
90	35.3	86.5	51.1
100	0.0	0.0	0.0

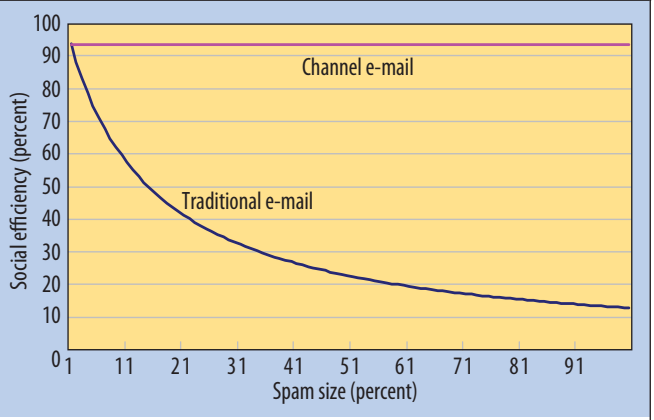


Figure 2. Social efficiency of traditional versus channel e-mail by spam size (spam level = 80 percent). While standard e-mail efficiency declines rapidly as spam size increases, channel e-mail efficiency is unaffected.

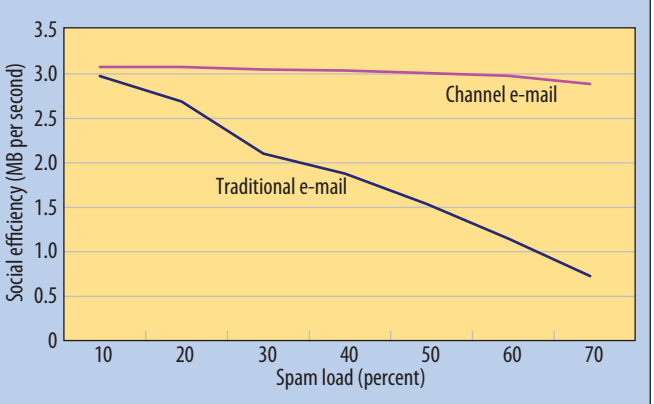


Figure 3. Social efficiency of traditional versus channel e-mail by spam rate for medium spam size. While standard e-mail performance dropped rapidly under spam assault, channel e-mail was more robust.

developments like image spam, which increase spam size, won't affect channel e-mail.

Simulated social efficiency

To test the theory, we simulated a communication network to compare traditional and channel e-mail. One computer sent messages to another over a local network isolated from outside influences including the Internet. In traditional mode, messages were just sent, but in channel mode, messages required channel permissions. A third computer simulated an outside spam source, using small (5 Kbytes), medium (10 Kbytes), and large (60 Kbytes) message sizes, at spam rates of 10 to 70 percent of the nonspam messages.

To estimate social efficiency, we directly measured nonspam message transmission time as the message size was known, giving an Mbytes-per-second rate. The rate was higher if nonspam messages arrived quickly and lower if they took longer due to network resources used by spam.

Figure 3 shows the social efficiency of traditional versus channel e-mail for medium-size spam messages. As predicted, increasing spam load drastically reduced social performance for traditional e-mail, but channel e-mail was much more robust. Again, increasing spam size drastically affected traditional e-mail but had little effect on channel e-mail.

Usability

To be accepted a new system must be usable as well as efficient. To compare the usability of channel versus traditional e-mail, we developed two matching Web-based e-mail prototype interfaces. The traditional interface had an inbox of messages received and an outbox of messages sent. The channel e-mail interface showed channel requests, sent e-mails, and current channels. Current channels could be open (known), closed (spam), and not yet classified.

In the channel prototype, e-mails from first-time senders went into the channel requests area, where the user could choose to "accept sender" (move to open channel area) or "reject sender" (move to spam area). Users could later move these e-mails again to another area—for example, to unclassified.

Because our study evaluated user acceptance, not network performance, the prototype didn't actually use a three-step channel send protocol but just sent simple messages. Subjects were in groups of 10, each with an allocated e-mail ID. Their task was to send 17 simple e-mail questions like "What is your birthday?" and also to respond to 17 such questions from other participants. The average time to complete the task was 43.4 minutes. In addition, all participants received incoming spam at rates of 12, 40, or 73 percent of valid e-mails sent.

The experiment randomly allocated subjects into two groups, one with the traditional e-mail prototype and the other with the matching channel e-mail prototype. Each group completed the task for three spam levels, then responded to five statements that each corresponded to one of four usability dimensions—understandability, learnability, operability, and perceived usefulness—from a previously validated model.[19] The statements were as follows:

- "I would find it easy to get this e-mail system to do what I want it to do." (Understandability)
- "To learn to operate this e-mail system would be easy for me." (Learnability)
- "I would find this e-mail system to be flexible to interact with." (Operability)
- "Using this e-mail system in my job would enable me to accomplish tasks more quickly." (Perceived usefulness)
- "Using this e-mail system would make it easier to do what I want to do." (Perceived usefulness)

The responses were on a seven-point scale: extremely (un)likely, quite (un)likely, slightly (un)likely, and neither likely nor unlikely.

As Figure 4 shows, the channel e-mail interface on average over all spam levels rated higher than the traditional e-mail interface on all four dimensions, and a t-test comparison of mean response scores was significant at the 0.01 level. Under spam load, users preferred to manage e-mail by conversation channels rather than by individual messages. To an engineer the "polite" requests of channel e-mail may seem inefficient, but over time and under a growing spam load social communication may in fact be more efficient.

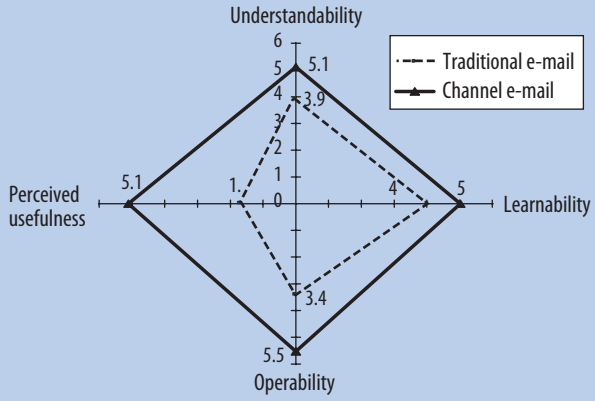


Figure 4. Usability of traditional versus channel e-mail interface. The channel e-mail interface rated higher on all four dimensions, and a t-test comparison of the mean response scores was significant at the 0.01 level.

IMPLEMENTATION

Challenge systems already implement a three-step send-challenge-resend protocol over the existing e-mail system, so implementing channel e-mail over current protocols is feasible. First-time senders must respond to challenges, then resend their e-mail. In contrast, channel e-mail seeks no "human proof" but simply asks senders to press reply to minimal-size e-mail request pings like,"Can we converse about <topic>? Press reply to receive my message."

Channel permissions could use channel properties like sender IP address plus request receive date/time, and could occupy one or several e-mail fields. User-generated tags, visible in the title, could let permissions be shared—for example, a website statement "Use e-mail tag happy-valley99" could direct readers to a preset open e-mail channel. Receivers could define incoming e-mail channels in advance—for example, the IT400 class could be told to use the permission tag "IT400" so all e-mails with that tag automatically sort into that channel. Compare this with the current situation, where e-mails flood into a single overflowing inbox, unless users set complex category filters.

Channel e-mail receivers could require senders to categorize their unsolicited e-mails into designated channels: "Please select a channel by putting one of these tag codes in your title: [mycompany], [myname], or [myhobby]." Note this isn't designed to be secure but dynamic and interactive. The system is flexible, as people can set tags to any needed complexity, including encryption. It's also dynamic, as closing and reopening a channel can create new permissions if the old ones are compromised or misused. Human social behavior can defend against mechanical spamming. The design options are beyond what can be outlined here, as users can create, delete, open, close, split, merge, and even transfer channels, with options like group and public-key channels.

DEPLOYMENT

For channel e-mail to evolve, it must survive a period of incomplete use. Two compatibility cases apply:

- *Nonchannel sender.* Channel e-mail treats full e-mails from new senders as channel requests and issues responses like, "Channel opened: Reply to this message to use the channel." People will reply as they know that relations take effort. Such messages may also recommend using channel e-mail to automate this step and explain why spam makes using channels necessary.
- *Nonchannel receiver:* In this case, a first time e-mail to someone else appears as a polite request: "XYZ wants to send you an e-mail on <topic>. Press reply to open a channel and receive the e-mail." Again the content could explain how channel e-mail works, and that this

request will only occur once per channel. Channel e-mail would automatically recognize any reply and send the pending message immediately.

That channel e-mail requires more effort to build connections is the cost of combating anonymous spam. While spammers take no care, making receivers bear all responsibility, channel e-mail senders must work to open a channel—though no more than cell phone users already do. This social cost is better than a global e-mail charge, as it primarily applies to unknown others. Users converting to channel e-mail can minimize start-up costs by automatically opening channels to people in their address book.

That to be received one must oneself receive is a social requirement of channel e-mail.

A channel e-mail advantage is that it reduces inbox numbers, as each visible line is a conversation of many messages. Gmail organizes messages like this, by conversation thread prioritized by last message recency (plus topic for some reason). Such threading keeps messages in the same conversation together, and avoids confusing trips between inbox and sent mail to figure out who said what. Unlike *topic-threading*, where users must maintain topic folders, in *conversation-threading* social acts—who you talk to—defines the threads. It's also harder to "lose" an e-mail, as it sits in a conversation context—the user would have to "forget" the entire conversation. Thread visualization can also be based on who replies to what.[20]

EFFECTIVENESS

The "Accept all" default means channel e-mail sends permissions to all unknown senders, including spammers. For people using channel e-mail, this is just part of the invisible handshaking of message sending. However, having to reply to an e-mail to send an e-mail presents spammers with a problem. If they don't receive and respond, then *none* of their e-mails are seen—that spammers send but don't receive is the perfect spam filter. Conversely, if spammers adopt channel e-mail to blend in, this reduces transmitted spam and allows channel closures. Also, receiving e-mail makes them vulnerable to user feedback and counter-spamming—for example, a proposed e-mail "Return to sender" button, which deletes the message and returns it to the sender with a "Return to sender:" title.[8] That to be received one must oneself receive is a social requirement of channel e-mail.

Channel e-mail democratizes black and white lists by letting users define who they want to talk to. People naturally do this, and systems like Facebook where individuals manage friend lists work well. While ISP lists are largely blacklists of spammers, channel e-mail lists will be largely white lists of known contacts. Spammers can create new identities to bypass blacklists, but for white lists they are still "unknown."

Like graylisting, channel e-mail suspects unknown e-mails, but rather than passively waiting some hours to respond, it replies immediately. It exploits the fact that most spammers never respond to e-mails to avoid being spammed themselves. Rather than first-time senders facing human test questions that smart software can solve anyway, the challenge is simply to reply.

Channel e-mail also works for community or organizational spam, which thwarts normal spam filters. In commercial spam, a few professional spammers send millions of spam messages, but in community spam, many people send selfish messages to everyone, like, "My daughter needs a piano tutor. Can anyone recommend one?" Community spam is as big a problem as commercial spam, as while one can block a spammer, blocking a community list is less desirable.

Channel e-mail offers other options: return the message or close the channel, creating an e-mail like "The user has closed this channel—to reopen. …" The social effect on those who routinely spam community lists could be dramatic. While commercial spammers usually don't receive replies, community spammers do—in spamming a channel e-mail community, they could the next day find their inbox filled with return-to-sender or channel-closure replies. Such community feedback is important. It lets people know how others see their acts, keeping a social balance between sender and receiver needs.

In summary, channel e-mail is

- *holistic*—it affects the whole system, not just cosmetic personal inbox effects;
- *democratic*—it distributes rather than centralizes control;
- *scalable*—as more people use it, more people manage their channels;
- *usable*—conversational threading involves fewer lines and less cognitive overload;
- *flexible*—it works for commercial and community spammers;
- *evolutionary*—it handles cases where other sender/receivers don't use it; and
- *transparent*—communication actions and responses are visible.

In channel e-mail, politeness isn't just nice, it's crucial to a higher social level of performance.[21]

Some spam researchers feel that smart filters are "holding the line" and that "we" will defeat "them" in the spam wars,[9] but for the rest of us, the Internet commons as a spam battleground is a losing proposition. Simple arithmetic shows that we can't outrun the spam challenge: If each of the 23 million businesses in America alone sends just one unsolicited message per year to all users, this equates to 63,000 e-mail messages per person per day. The spam potential is the square of the number of users, and in a future with billions online, it easily outstrips Moore's law of technology growth.

If e-mail "dies" from spam infestation, don't imagine that like slash-and-burn farmers we can simply move on to new pastures. We carry the spam plague with us, in socially primitive software designs, so it will simply infect new applications, as SPIM and SPAT illustrate. Indeed, spam is just the poster boy for a whole genre of antisocial acts that threaten online society, including spyware, phishing, spoofing, scams, unwanted pornography, identity theft, libel, privacy invasions, piracy, plagiarism, and online harassment.

Spam is a social dilemma inherent to social interaction. It simply won't go away until it's addressed. The problem isn't a flaw in human nature—just a matter of getting rid of the "bad guys." Without the drive for individual gain, humanity could never have evolved, just as without a sense of community, we couldn't have become civilized. That some people will be selfish, with their survival drive overriding their social drive, is inevitable given our origins. Consequently, societies have for thousands of years found ways to keep the problem in check. The only difference now is that we must make our social principles explicit, to implement them in code. Computer engineers once needed to know only technology to design technical systems. Now they must understand social concepts as well, to design sociotechnical systems. While this seems a lot to ask, the gain is that sociotechnical systems can literally change the world.

It would be naïve to expect to introduce a new form into a system as complex as global e-mail without problems—for example, security add-ons might be needed to combat phishing attacks. Yet more of the same isn't an option if it inevitably leads to a 99 percent spam-clogged end point where everyone loses, including spammers.

The social problems of technical systems can't be ignored forever. Spam is the social equivalent of cancer, which selfishly grows until it kills its host. Only changing from technical to social requirements can unlock higher social software performance levels.[22] While channel e-mail initially seems less efficient, as the spam reality bites it quickly becomes much more efficient.

This may be why sociability evolved—because it works. Pure technology and pure socially based responses have had their chance. Simple one-step, one-way messaging systems that ignore the mutual rights of conversation will become a thing of the past. The future will require a sociotechnical approach to sociotechnical problems like spam.

Acknowledgments

Thanks to Victor Bi and Zheng Dai for the channel e-mail interface and network simulation work.

References

1. Messaging Anti-Abuse Working Group, "*E-mail Metrics Program: The Network Operators' Perspective*," report no. 2, June 2006; www.maawg.org/about/FINAL_1Q2006_Metrics_Report.pdf.
2. S.J. Vaughan-Nichols, "Saving Private E-mail," *IEEE Spectrum*, Aug. 2003, pp. 40-44.
3. A. Weiss, "Ending Spam's Free Ride," *netWorker*, June 2003, pp. 18-24.
4. P. Boutin, "Can E-mail Be Saved?" *InfoWorld*, 16 Apr. 2004; www.infoworld.com/article/04/04/16/16FEfuturemail_1.html.
5. Metrics 2.0, "2006: The Year Spam Raised Its Game; 2007 Predictions," 18 Dec. 2006; www.metrics2.com/blog/2006/12/18/2006_the_year_spam_raised_its_game_2007_prediction.html.
6. Nucleus Research, "Spam: The Serial ROI Killer," research note E50, June 2004; http://cnscenter.future.co.kr/resource/security/application/e50.pdf.
7. Ferris Research, "Reducing the $50 Billion Global Spam Bill," 22 Feb. 2005; www.ferris.com/?page_id=73258.
8. B. Whitworth and E. Whitworth, "Reducing Spam by Closing the Social-Technical Gap," *Computer*, Oct. 2004, pp. 38-45; http://brianwhitworth.com/spam-computer.pdf.
9. J. Goodman, G.V. Cormack, and D. Heckerman, "Spam and the Ongoing Battle for the Inbox," *Comm. ACM*, Feb. 2007, pp. 24-33.
10. L. Weinstein, "Inside Risks: Spam Wars," *Comm. ACM*, Aug. 2003, p. 136.
11. E. Harris, "The Next Step in the Spam Control War: Greylisting," white paper, rev. 21 Aug. 2003; http://projects.puremagic.com/greylisting/whitepaper.html.
12. R. Axelrod and W.D. Hamilton, "The Evolution of Cooperation," *Science*, 27 Mar. 1981, pp. 1390-1396.
13. M. Bazeley, "New Weapon for Spam: Bounty," *San Jose Mercury News*, 26 Apr. 2003.
14. B. Whitworth and A. de Moor, "Legitimate by Design: Towards Trusted Socio-Technical Systems," *Behaviour & Information Technology*, Jan.-Feb. 2003, pp. 31-51; http://brianwhitworth.com/legitimacy2002.pdf.
15. C. Dwork and M. Naor, "Pricing via Processing or Combatting Junk Mail," *Advances in Cryptology—Crypto 92*, LNCS 740, Springer Verlag, 1993, pp. 139-147.
16. L. Lessig, *Code and Other Laws of Cyberspace*, Basic Books, 1999.

17. B. Whitworth, "Polite Computing" *Int'l J. Virtual Communities and Social Networking*, April-June, 2009, pp. 65-84; http://brianwhitworth.com/polite05.pdf.
18. Z. Duan, Y. Dong, and K. Gopalan, "A Differentiated Message Delivery Architecture to Control Spam," *Proc. 11th Int'l Conf. Parallel and Distributed Systems—Workshops* (ICPADS 05), vol. 2, IEEE CS Press, 2005, pp. 255-259.
19. N. Bevan, "Quality and Usability: A New Framework," *Achieving Software Product Quality*, E. van Veenendaal and J. McMullan, eds., Uitgeverij Tutein Nolthenius, 1997, pp. 25-34.
20. B. Kerr, "Thread Arcs: An Email Thread Visualization," *Proc. 2003 IEEE Symp. Information Visualization* (InfoVis 03), IEEE Press, 2003, pp. 211-218.
21. B. Whitworth, J. Fjermestad, and E. Mahinda, "The Web of System Performance," *Comm. ACM*, May 2006, pp. 93-99; http://brianwhitworth.com/wosp06.pdf.
22. B. Whitworth "The Social Requirements of Technical Systems," *Handbook of Research on Socio-Technical Design and Social Networking Systems*, B. Whitworth and A. de Moor, eds., IGI, 2009; http://brianwhitworth.com/STS/STS-chapter1.pdf.

Brian Whitworth *is a senior lecturer at Massey University—Albany, Auckland, New Zealand. His research interests include sociotechnical design, evaluation, and operation. Whitworth received a PhD in information systems from the University of Waikato, Hamilton, New Zealand. The Channel E-mail project is currently a design, not an implementation. Contact him at bwhitworth@acm.org.*

Tong Liu *is a tutor at Massey University—Albany. Her research interest focuses on designing online systems to better support human and social interactions. Liu received an MS in computer science from Massey University—Albany. Contact her at t.liu@massey.ac.nz.*