Email dependability

Tim Moors
School of Electrical Engineering and Telecom.
University of New South Wales
Sydney, NSW, Australia
http://www.eet.unsw.edu.au/~timm
Importance of email

>0.5B users of business email worldwide\(^1\)
Tens of billions of messages sent daily
  (60B, 30%+ spam, by 2006)\(^1\)
Top message sources\(^2\):
  • 300M/day: Yahoo
  • >100M messages/day: 13+ service providers

Comparable systems:
  • Australia Post\(^3\): 5B items p.a.
    • Reliability is required by law, and audited.
    • For domestic letters: 96.5% on time or early
      98.9% less than one day late.
      What about the remaining 1.1%?!
  • Short Message Service (SMS):
    • Typical day for an Australian telco SMS Center\(^4\):
      1.2M messages delivered (0.5M after temporary failure),
      10k (1%) abandoned
    • 7.5% of SMS messages lost between carriers in US in
      Dec. 2002\(^5\)

How reliable are email systems?

1. IDC: *Worldwide Email Usage Forecast, 2003-2007*
2. www.senderbase.org
3. Australia Post: *Annual Report, 2003*
4. Personal communication
5. Keynote *Study of Wireless Text Messaging, 2003*
• Importance of email
• Email features affecting dependability
  • Asynchrony
  • Indirection
• Email outcomes
  • Positive / negative
  • Difficulties with notifications
• Contributing factors
  • Degeneration of email
  • Sources of errors
• Measurement study
• Tools for enhancing dependability
Email features 1: Asynchronous

Available?  
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
- Yes
- No
- No
- Yes
Mail servers are *entrusted* with responsibility to forward messages.

“the protocol requires that a server accept responsibility for either delivering a message or properly reporting the failure to do so.” [RFC 2822]

- Nothing protects against server failure, e.g. crashing or overload. Server can lose messages.
- End-users can’t determine the trustworthiness of a remote server.

Mail servers may *persist* in attempting to deliver messages

- Persistence is unpredictable since it varies with the server e.g. Sendmail warns after 4 hours and persists for 5 days
- Server persistence may delay source awareness of problems
- Failure notifications may be tardy, e.g. 1 month!
Email features 2: Indirection

• An email address indicates where a message will go next.
• The message may then be forwarded elsewhere. e.g. aliases (tim@trusty.com → tim@cheap.com) mail that is redirected (boss@eg.com → worker@eg.com) mail to a mailing list
• A sender may be oblivious to identity of ultimate recipient(s).
  • If a sender expects a delivery notification, then who does it expect it from?
  • Notification from next address doesn’t imply message has reached ultimate destination(s) and had desired impact.
• Receiver’s delivery concerns extend beyond those of the source. ⇒ Need receiver-oriented mechanisms as well as source-oriented mechanisms.
Spectrum of possible email outcomes

<table>
<thead>
<tr>
<th>Action</th>
<th>Acknowledgement</th>
<th>No loss</th>
<th>But still trying</th>
<th>Parties are aware</th>
<th>Silent discard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delivered</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not delivered</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Delay = quality

None → Extensive delay

From: Mail Delivery Subsystem
Subject: **Warning**: could not send message for past 4 hours
The original message was received at ...
from foo.com
----- The following addresses had transient non-fatal errors
  <a@b.com>
----- Transcript of session follows -----  
<a@b.com>... Deferred: Connection reset by example.com.
Warning: message still undelivered after 4 hours
Will keep trying until message is 5 days old

From: Mail Delivery Subsystem
Subject: **Returned mail**: see transcript for details
The original message was received at ...
from foo.com
----- The following addresses had permanent fatal errors
  <a@b.com>
----- Transcript of session follows -----  
550 5.1.2 <a@b.com>... Host unknown ...
Notification challenges

Events of interest:
- delivery (Delivery Service Notifications – RFC 3461-4)
- disposition (Message Disposition Notifications – RFC 3798)
  aka “read receipts” “acknowledgements” etc

Notifications may be positive, negative, or missing

Challenges:
- Notification may come from intermediary, not end-system.
- End-system disposition is what matters, but impinges on privacy.
- What does “received” mean? Downloaded, skim read, acted on?
- Notifications are not delivered reliably ⇒ Lack of notification proves nothing.
- Scalability: Source can be inundated with positive notifications from many mailing list recipients
Degeneration of email

Malicious worms and nefarious spammers cause:

**Discard of messages**
- **by filters.** False detection rate is:
  - Low for virus/worm filters
  - Substantial for spam filters, particularly when applied to commercial messages
- **by overflow** of storage systems:
  Spam/worms fills mailboxes/servers and blocks legitimate email

**Users ignore failure notifications**
- Users get flooded with failure notifications about messages they didn’t send.
- **Forged notifications:** Sent by worms (spammers next?) to entice users to open infected messages.
  e.g. by Netsky, Evaman, Klez, Mydoom
- Genuine notifications in **response to forged messages:**
  Spammers and worms harvest addresses and use them to send forged email. Users receive non-delivery notifications.
- Appropriate client software **could** filter out unsolicited notifications.
Sources of errors

1. Misaddressing by source
e.g. tim@trusty.com vs tom@trysty.com

2. Improper sending by source
e.g. stuck in drafts folder

3. Discard by servers, e.g. due to
   • storage overflow
   • crashes
   • buggy implementation
   • ...

4. Filtering of spam/viruses
   (Of particular concern for messages with commercial content.)

5. Incorrect download
   e.g. inadvertent deletion

Our initial focus
Email measurements

Periodically send innocuous probe messages to accounts with several mail providers:
- 2 large portals (Yahoo, Hotmail)
- 2 ISPs (Bigpond, Optusnet)
- 2 universities
- 12 free mail providers

Each probe includes a sequence number and time stamps enabling measurement of loss/duplication and delay.

Each provider lost some messages during the experiment. Overall, 1.57% lost, about 0.5% lost silently. Loss rate as high as 10% over a month for one provider.

† Work done with Anthony Lang, 4th year engineering student
6 months of measurements

Loss: 3 silent (650ppm) + 15 notified (0.3%)
59 delay warnings (1.3%), 3 duplicates (650ppm)
(4630 messages in total)
Delay of all messages

Email feature 3: Source usually can’t measure performance (delay) because of lack of feedback

92% of all messages delayed < 30 sec
Our work 1: Measure and store

- **Measurement of:**
  - email delays, loss, duplication
  - server availability
    - reliability
    - need for asynchronous protocols
  - prevalence of servers with alternate MX records

- Massive (TB) storage systems (e.g. for email) based on peer-to-peer “file sharing” technology (robust and growable)
Our work 2: Client add-ins

New tools to help users cope with unreliability
Implemented as mail client add-ins and proxies
Assuming we can’t change the world of mail servers overnight.

At sources:
• **Automatically retransmit mail** to backup accounts in case primary account fails (effectively X.400 “alternate recipients”)
• Match incoming to outgoing email
  • Highlight outgoing mail that hasn’t been responded to ==? lost (Using Message-IDs and In-Reply-To: fields)
  • Filter out notifications regarding messages client did not send
• **SMTP traceroute**: Allow a source to determine which servers a message passes through, suggesting possible failure points

At destinations:
• Trails of message meta-information (e.g. time, source/destination) in case of accidental message deletion
• Sequence numbers to detect loss, particularly for mailing lists
• Real-time monitoring and notification of transfer delays
Conclusions

- Email *does* get lost. Sometimes (0.1%?) silently.
- Store-and-forward email (even with notifications) fundamentally *can’t* guarantee delivery.
- New tools can enable end-users to cope with unreliability.

For more information, see
http://uluru.ee.unsw.edu.au/~tim/dependable/email
or try your luck emailing t.moors@unsw.edu.au

We’re looking for collaborators.