# Scribee Experimentation – Early Statistics on Email Conversations

Vincent Verdot, Vincent Toubiana, [TBC]
Hybrid Communications dpt.
Alcatel-Lucent Bell Labs – Application domain
Nozay, France
*{firstname.lastname}*@alcatel-lucent.com

***Abstract*—Information overload is a real challenge becoming more and more critical regarding the quite limited human capabilities. 107 trillion emails sent in 2010; that is more that we could even imagine so how to help the user to take advantage from it? That is the question we addressed in Scribee, a multidisciplinary internal Bell Labs initiative which focuses on the very pragmatic case of email service. From a knowledge and conversation-based approach we try to improve the email technologies in the context of information overload. Scribee itself did not yet provided its conclusions but the experiment offers interesting statistics on email: half of messages' content is useless, most conversations ends within a business week, emails are not equal… This analysis based on real email conversations provided valuable information about this old communication tool and could help improving it.**

***email; conversation; communication; information overload***

## I. INTRODUCTION

"Information overload", or how too much information kills information, is today one of the major challenges the IT sphere has to address. Supported for the last 15 years by the improvement of network technologies, the success of Internet and the Web 2.0 style based on users' generated contents, the amount of exchanged information has increased exponentially.

With a Worldwide Internet that initially consisted of 39.6 million users in 1995, 361 million in 2000 and finally 1,967 million (2010) in a Web that promotes users' creation and sharing, the share of data generated everyday has become so huge that it cannot be handled at a human scale. Apart from possible technical limitations, it is mostly a human scale problem… which amount of data is the brain able to transform/acquire into knowledge? Some numbers to understand the phenomenon: 152 million of blogs [1], 5 billion photos hosted by Flickr, 2 billion video watched per day on Youtube, 110 million tweets sent per day… how to allow the user to get benefit from that?

We focus our research works on communication technologies and the impact of information overload on them. One of our previous activities was dedicated to the email service. Indeed, even if you are not an active Web user, avoiding any social or media websites, information overload still comes to you through your email box. The electronic mail is significantly concerned; in 2010 we counted about 107 trillion emails sent (only for that year), that is, 294 billion messages per day. If you exclude *spam* (automatically sent unsolicited messages) which represents about 89.1%, it still remains 32 billion messages sent worldwide every single day.

We adopted a very pragmatic approach, trying to redesign the way people interact with their email client software, and looking for what they actually expect from it: capturing conversations, finding relevant information, transforming into knowledge. Google Wave [2] somehow addressed this issue but we preferred to stay focused on the email technology instead of proposing yet another communication service.

So our strategy is definitely knowledge-oriented, considering conversations as a whole instead of each single piece of message (typically an email); in this context we started one year ago a project called *Scribee*. The "internal" nature of this project makes it very little documented, moreover it is not yet over and so the final conclusions will be provided later. However we conducted an experimentation that produced interesting statistics on email conversations which are presented in this document.

The remaining of this article will focus on presenting the early results of the Scribee experimentation, relative to email-based conversations. In Section II we will briefly describe the experimentation and its context, emphasizing on the pragmatic aspects of this study. Then in Section III we will show a selection of statistics we produced, providing a short analysis for each of them. Finally we will conclude on the obtained results and the future perspectives.

## II. EXPERIMENTATION CONTEXT

### A. Overview

Scribee is an internal multidisciplinary project which brings together researchers, engineers, designers, sociologists… around a key objective: building knowledge from email conversations; or how to build efficient email client software in an information overload context? In this framework, we ran a wide experimentation among Bell Labs researchers in order to analyze email usages and characteristics but also to evaluate our prototype and mechanisms.

As mentioned in introduction, the project itself is not yet over and so the final results are not ready to be published, nevertheless we did a number of interesting statistics and analysis on email conversations that deserve to be shared.

### B. Participants

The experiment was open within the Bell Labs and all participants were volunteers (158 joined). The prototype did

not required any specific installation, users just had to add a specific address (*e.g.* scribee@bell-labs.com) to the email's destination (To, Cc, or BCc fields) every time they wanted to use the prototype. This one was working transparently (seamless for senders and receivers) and very easy to use, so it greatly facilitated its adoption for a "normal" use, leading to more relevant results.

Participants' profiles were heterogeneous but all were familiar with computers and communication software, using them every day. Researchers and engineers participating to the project also actively used the prototype for collaboration. This inferred significant email traffic; however it was very valuable regarding our study that is focused on structured conversations.

### C. Duration

The data presented in this document corresponds to the measures we realized during 6 months. As the participants were free to use or not the prototype for every message they sent, the recorded activity significantly fluctuated during the (quite long) experiment and so for various reasons: popularity of the project, updates on the prototype, etc.

### D. Data gathering process

We collected and recorded every email messages sent to our prototype, which was finally just an additional recipient for all conversations. The received messages were processed, updating various counters (number of messages, etc) and then anonymously stored as conversation trees for "offline" analysis (log files). Building conversations was mainly based on unique identifiers present in the email headers.

Advanced analysis was performed later, offline, by parsing the log files, allowing for example to produce statistics such as the email payload, conversation length, etc. The results are presented in the following section.

## III. RESULTS

### A. General Information

The Scribee experimentation allowed us to study and analyze email conversations in basic (office) situations, involving real email interactions. However, note that the data was collected in experimental conditions, and the obtained results may differ from real conditions, so they must be used knowingly. As indicated in the previous section, the use of our prototype by the participants was optional so the data sample is relatively small regarding the duration of the experiment. Also we will not detail some statistics relying on the extensive usage of the service (*e.g.* number of emails per user, etc) but focus on trend and time-independent values. Nevertheless, the results bring a valuable outlook on email conversations and their characteristics.

Before to go into advanced analysis, here are some numbers. During 6 months, we collected 601 emails organized in 303 distinct conversations and which involved 158 users. After some "cleaning" (removed few trivial debug messages), we only considered 246 conversations for further processing. Even if it may present a weak average activity, conversations are correctly logged as the inclusion of Scribee address in at least one message is enough to contaminate the whole conversation (replying to a conversation includes the previous recipients).

### B. Messages

The structure of a single email message is defined by the Simple Mail Transfer Protocol (RFC 5321 [3]) and consists of two different parts: a header and a body. The header holds various data useful for the servers and clients assuring the email service (metadata). The body is typically what is displayed by the client software; that is the actual email content. Our statistics are only based on the body part of messages.

The Figure 1 describes the content of collected data (601 messages), considering the conversation context. The *payload* (49%) is the new information of a message, *i.e.* the data which is added to a conversation by sending the email. This is the actual true payload of an email. The *redundant data* is the data carried by the message but which already exist within the conversation (we detected redundancy thanks to a *diff* algorithm). This is typically the body of the email you just replied to (and which is copied in the new email body). Finally, the *email client headers* are those text separators automatically added by the client between messages (within a single email), indicating the original message's sender, recipient, etc.

This first result shows that half of an email is useless. Maybe not truly useless (reminds the context, support for inline comments, etc) but considering that client softwares are able and actually do track the conversations and the emails context, we could wonder if such inefficient redundancy is really required. Moreover, this redundant data is sometimes hidden by the client itself (*e.g.* folded text in Gmail [4]).

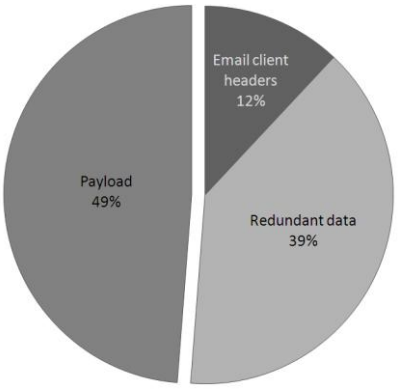


Figure 1. Characteristics of collected data

Redundant data is automatically added by the client into the email body when replying to a message. Naturally, after several consecutive replies, the useless data stack up; consequently every new message will hold less and less valuable information. The share of the actual payload of an email within a conversation can be expressed as follows.

$$P_n = \frac{S_n}{\sum_{j=1}^{n}(S_j + \mathrm{H_j})}$$

With $P_n$ the payload of the n[th] message (share of the message total size), $S_n$ the size of the n[th] message and $\mathrm{H_n}$ the size of the header of the n[th] message.

Thus, we can draw the graph presented in Figure 2 which shows the percent of useful information hold by an email within a conversation (the horizontal axis indicates the n[th] message within the conversation). We considered here that the email clients always added the redundant body information (actually quite usual) and that the header and message sizes are respectively 300 and 1000 bytes (according to what we measured on average).

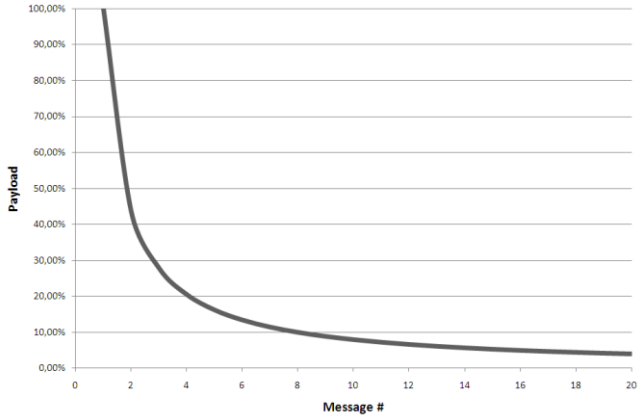


Figure 2. Message payload regarding rank in a conversation

The graph shows that the payload a message dramatically decreases after a few replies. At the fifth level (so at the fourth reply), the payload is only 15% of the message. As the conversation is likely to involve even more interactions, we can easily say that the model looks totally inefficient, and regarding the amount of email exchanged worldwide every day, the amount of data uselessly transferred is simply huge.

### C. *Conversations*

A conversation is a sequence of messages linked each others thanks to unique identifiers hold by the header part of the email and defined in SMTP. These identifiers are handled by the email client when the user explicitly decides to answer to a message. So, potentially a conversation (usually involving several same people around a common topic) may sometimes be broken when a link is lost (missed email, delivery failure, etc) or if a user manually creates a new message without using the built-in "reply-to" function of his client software. In Scribee we implemented specific mechanisms to cope with these issues, mostly for the former case; in other situations we were assuring a best effort mode.

The Figure 3 shows a timeline of the conversations we recorded (246 exactly, they kept their initial index so the max is 300 but there are gaps around the 250[th]). Each conversation is displayed vertically, with little squares representing every message, vertically aligned for a given conversation. On the vertical axis is indicated the time of emission for every email.

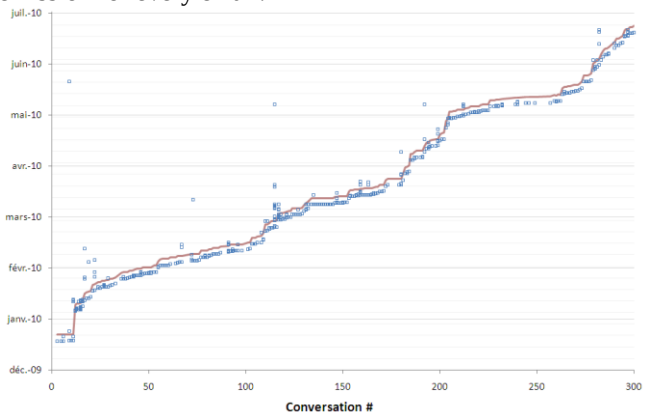


Figure 3. Conversations timeline

Although this chart does not bring much information, it clearly shows the email activity over the 6 months experiment. The higher the plain line slope is, the less activity occurred (time is on the vertical axis). This plain line which seems to closely follow the squared plots represents the 3[rd] quartile of average conversation duration. We will detail it later.

The Figures 4 presents the distribution of the conversations (246) regarding the number of sent messages. On the horizontal axis is indicated the number messages from 1 to 15+ and on the vertical axis the number of related conversations.

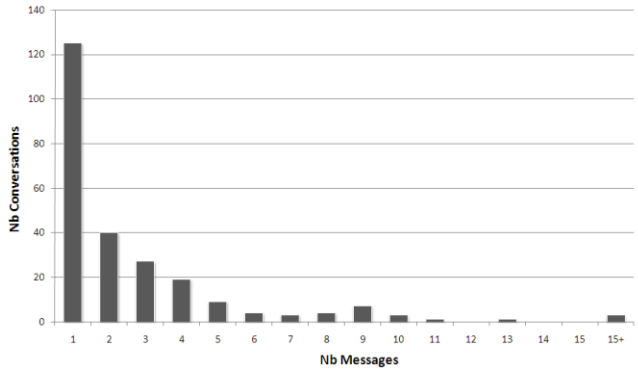


Figure 4. Distribution of conversations per number of messages

Almost all the conversations own 10 or less messages and very few count more than 15, which looks like reasonable values; long conversations are sporadic. However we also notice that a great number of conversations (120+) only consist of 1 message. Regarding our definition these cannot actually be considered as "conversations" because there is no interaction between the users. We will call them "notifications" or "announcements" as they consist in unidirectional data pushes, typically to notify users about some information. We can explain the high presence of these

announcements as Scribee was used by the project managers who regularly sent notifications (bugs, new releases, tests, etc). However, this symptom clearly state that the email box may hold different types of messages: notifications, conversations… Is it so obvious to treat them the same way? Do they all carry knowledge? Should they be archived or deleted after some time? Such questions will not be addressed in this article but they could lead to new perspectives for email technologies.

Figure 5 shows the distribution of conversations regarding the number of involved users. You may notice that conversations involving only 1 user are not displayed as not applicable (actually it initially existed a few for debug purpose). We can see that conversations with less than 8 users represent the wide majority of them, the same applied (less clearly) for conversations up to 5 emails. These thresholds (8 users / 5 messages) may be used as default values in various applications (queues, buffers, displays) as they will fit in most situations.

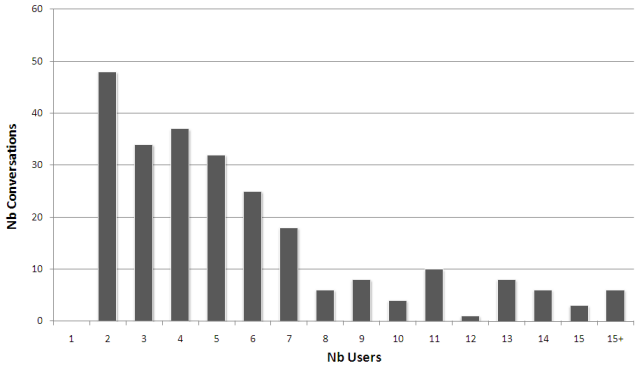


Figure 5. Distribution of conversations per number of users

### D. Conversation duration

By parsing the logged data, we also computed the length of conversations. To get the duration, we need the beginning and ending timestamps; if the start date is easy to retrieve (first message), when is a conversation over? Actually a conversation never ends as anytime a participant may answer an email thread, even several years after the last activity.

As the experiment was 6 months long, we arbitrarily took the last message within the conversation the determinate its duration. If we take a look at the Figure 3, we can see that very few conversations were interrupted (if any). The results are presented as a box plot in Figure 5; the announcements were excluded from this chart as they are not proper conversations (and their duration would be null).

Figure 6 shows the distribution of conversations' duration over an exponential time scale expressed in hours. The median duration is about 15 hours which roughly represents 2 business days, so half of conversations end within that interval. The third quartile indicates that 75% of conversations end within 3.8 days, *i.e.* one business week. This upper quartile was drawn on Figure 3 graph (as a plain line). This threshold could be used as reference value in email clients for archiving, ending conversations, etc and so improve this communication tool.

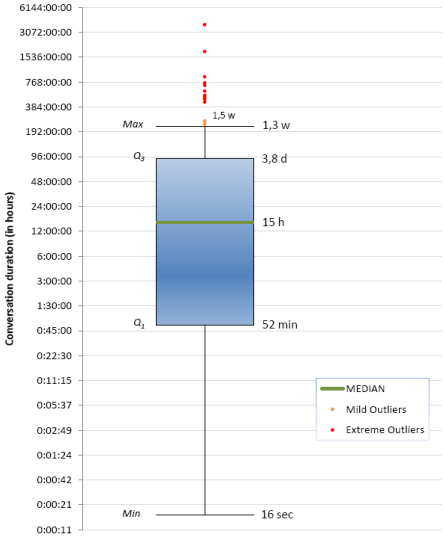


Figure 6. Conversation duration

## IV. Conclusion

Even if the Scribee project is not yet over, it allowed us, through an extensive and realistic experiment, to discover many aspects of email conversations. We were able to identify several issues in the management of electronic messages and the related conversations. We measured the payload of messages and the duration of conversations. We provided reference values that could be used as default threshold in various applications (for storage, display, etc).

Beyond acquiring and sharing this data, the Scribee project also addresses the email conversations as a knowledge management issue. We already included these recent results in the project in order to improve our approach and offer alternative solutions to the user facing the information overload big thing.

## Acknowledgment

We want to thanks all the members of Bell Labs Application Domain who actively participated in the Scribee project.

## References

[1] All numbers are from pingdom.com, we will detail the sources for the camera ready. "http://royal.pingdom.com/2011/01/12/internet-2010-in-numbers".

[2] Google Wave, "http://wave.google.com".

[3] J. Klensin, "Simple Mail Transfer Protocol," IETF, RFC 5321, October 2008.

[4] Gmail. "http://mail.google.com".