International Journal of Information and Computation Technology.
ISSN 0974-2239 Volume 3, Number 11 (2013), pp. 1187-1196
© International Research Publications House
http://www. irphouse.com /ijict.htm

The Undo Sent E-mail (USE) Protocol

Praveen.B – 4th year B.E (CSE)

Department of Computer Science Engineering, Sri Venkateswara College of Engineering, Chennai, India.

Abstract

This paper provides the sender of email the power to undo his sent mail. I provide our own idea in implementing this undo function in sending mails with my own protocol called the USE-Undo Sent E-mail. This is achieved by adding an extra functionality to the domain's SMTP server and the message can be pulled by both the sender and receiver from the POP3/IMAP server. If the sender pulls the mail first,then it means he has successfully retrieved the mail and if the receiver pulls the mail first,then it means the receiver has read the mail and the mail cannot be retrieved. I achieve this by implementing the USE protocol.

Keywords: USE-Undo Sent E-mail Protocol; adding extra functionality to SMTP domain server; POP3/IMAP server;

1. Introduction

To err is human and as users we often make mistakes and the designers should always keep this in mind while designing. Undo is the ultimate safety net, lending an incredible sense of solidity to an interface. That's why every desktop application from Word to Photoshop provides multiple-level Undo. So, then, why E-mails that are sent once cannot be undone? The answer that we often get is that Undo is hard to implement in E-Mails beacause it is time-sensitive in nature. In sending E-mails,where we don't have the Undo option,before performing the send,a warning may be displayed.But the simple fact is **warning can never be a substitute for Undo**.A user may just ignore the warning and continue his action but later may regret.

In this paper,my goal is to give an idea about providing Undo functionality in sending E-Mail. **Adding Undo to your E-mail interfaces profoundly and positively affects the usability of your site**. It reduces user frustration, and increases user trust.

Both of those outcomes mean that more users use your mail service. Remember: To the user, **the interface is the product**.

Now in the beginning we will discuss about the general architecture.Later in the paper We give our idea of how we can implement the undo option in sendig e-mail.

2. The Push Technology in Email

Email is a push system: the SMTP protocol is a push protocol (see Push e-mail). However, the last step —from mail server to desktop computer— typically uses a pull protocol like POP3 or IMAP. Modern e-mail clients make this step seem instantaneous by repeatedly polling the mail server, frequently checking it for new mail.

Push email is used to describe email systems that provide an always-on capability, in which new email is actively transferred (pushed) as it arrives by the mail delivery agent (MDA) (commonly called mail server) to the mail user agent (MUA), also called the email client. Email clients include smart-phones and, less strictly, IMAP personal computer mail applications. Regardless of whether the receiver uses polling email, outgoing mail is generally *pushed* from the sender to the final mail delivery agent (and possibly via intermediate mail servers) using Simple Mail Transfer Protocol. However, if the receiver uses a polling email delivery protocol, the final step from the last mail delivery agent to the client is done using a poll. Post Office Protocol (POP3) is an example of a polling email delivery protocol. At login and later at intervals, the mail user agent (client) polls the mail delivery agent (server) to see if there is new mail, and if so downloads it to a mailbox on the user's computer. Extending the "push" to the last delivery step is what distinguishes push email from polling email systems.

The reason that polling is often used for the last stage of mail delivery is that, although the server mail delivery agent would normally be permanently connected to the network, it does not necessarily know how to locate the client mail user agent, which may only be connected occasionally and also change network address quite often. For example, a user with a laptop on a WiFi connection may be assigned different addresses from the network DHCP server periodically and have no persistent network name. When new mail arrives to the mail server, it does not know what address the client is currently assigned.

The Internet Message Access Protocol (IMAP) provides support for polling and notifications. When a client receives a notification from a server, the client may choose to fetch the new data from the server. This makes retrieval of new messages more flexible than a purely-push system, because the client can choose whether to download new message data

3. Email Protocols: IMAP, POP3, SMTP And HTTP

Basicaly, a protocol is about a standard method used at each end of a communication channel, in order to properly transmit information. In order to deal with your email you must use a mail client to access a mail server. The mail client and mail server can exchange information with each other using a variety of protocols.

3.1 IMAP Protocol

IMAP (Internet Message Access Protocol) – Is a standard protocol for accessing e-mail from your local server. IMAP is a client/server protocol in which e-mail is received and held for you by your Internet server. As this requires only a small data transfer this works well even over a slow connection such as a modem. Only if you request to read a specific email message will it be downloaded from the server. You can also create and manipulate folders or mailboxes on the server, delete messages etc.

3.2 POP3 Protocol

The **POP (Post Office Protocol 3)** protocol provides a simple, standardized way for users to access mailboxes and download messages to their computers.

When using the POP protocol all your eMail messages will be downloaded from the mail server to your local computer. You can choose to leave copies of your eMails on the server as well. The advantage is that once your messages are downloaded you can cut the internet connection and read your eMail at your leisure without incuring further communication costs. On the other hand you might have transferred a lot of message (including spam or viruses) in which you are not at all interested at this point.

3.3 SMTP Protocol

The **SMTP (Simple Mail Transfer Protocol)** protocol is used by the Mail Transfer Agent (MTA) to deliver your eMail to the recipient's mail server. The SMTP protocol can only be used to send emails, not to receive them. Depending on your network / ISP settings, you may only be able to use the SMTP protocol under certain conditions

3.4 HTTP Protocol

The HTTP protocol is not a protocol dedicated for email communications, but it can be used for accessing your mailbox. Also called web based email, this protocol can be used to compose or retrieve emails from an your account. Hotmail is a good example of using HTTP as an email protocol.

4. Mail Server

A mail server (also known as a *mail transfer agent* or MTA, a *mail transport agent*, a *mail router* or an *Internet mailer*) is an application that receives incoming e-mail from local users (people within the same domain) and remote senders and forwards outgoing e-mail for delivery. A computer dedicated to running such applications is also called a mail server. Eg: Microsoft Exchange, Gmail, . The mail server works in conjunction with other programs to make up what is sometimes referred to as a messaging system. A messaging system includes all the applications necessary to keep e-mail moving as it should. When you send an e-mail message, your e-mail program, such as Outlook or

Eudora, forwards the message to your mail server, which in turn forwards it either to another mail server or to a holding area on the same server called a *message store* to be forwarded later.

4.1 Types of Mail Servers

Mail servers can be broken down into two main categories: outgoing mail servers and incoming mail servers. Outgoing mail servers are known as **SMTP**, or Simple Mail Transfer Protocol, servers. Incoming mail servers come in two main varieties. **POP3**, or Post Office Protocol, version 3, servers are best known for storing sent and received messages on PCs' local hard drives. **IMAP**, or Internet Message Access Protocol, servers always store copies of messages on servers. Most POP3 servers can store messages on servers, too, which is a lot more convenient.

4.2 The IMAP/POP3 Server

What's the difference?

The main difference, as far as we are concerned here, is the way in which IMAP or POP controls your e-mail inbox.

When you use IMAP you are accessing your inbox on the U of M's central mail server. IMAP does not actually move messages onto your computer. You can think of an e-mail program using IMAP as a window to your messages on the server. Although the messages appear on your computer while you work with them, they remain on the central mail server.

POP does the opposite. Instead of just showing you what is in your inbox on the U's mail server, it checks the server for new messages, downloads all the new messages in your inbox onto your computer, and then deletes them from the server. This means that every time you use POP to view your new messages, they are no longer on the central mail server. Figure 1 illustrates these concepts.

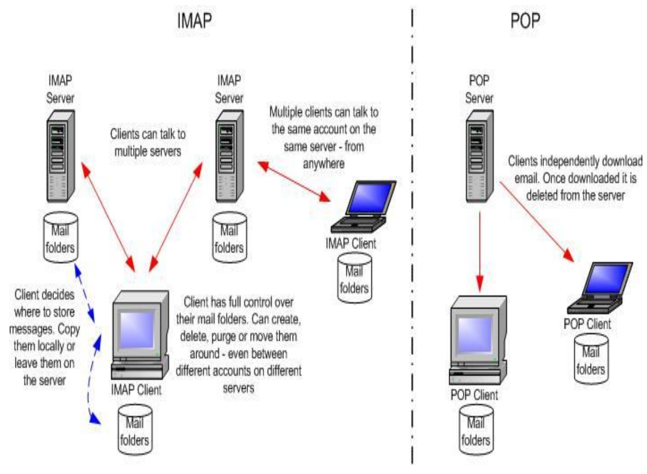

IMAP
POP
IMAP Server
IMAP Server
Clients can talk to multiple servers
Multiple clients can talk to the same account on the same server - from anywhere
POP Server
Clients independently download email. Once downloaded it is deleted from the server
Mail folders
Mail folders
IMAP Client
Mail folders
Client decides where to store messages. Copy them locally or leave them on the server
Client has full control over their mail folders. Can create, delete, purge or move them around - even between different accounts on different servers
IMAP Client
Mail folders
POP Client
Mail folders
POP Client
Mail folders

Fig. 1 (a): The functions performed in POP3/IMAP server.

5. The Process of Sending an Email

Now that you know the basics about incoming and outgoing mail servers, it will be easier to understand the role that they play in the emailing process. The basic steps of this process are outlined below for your convenience.

Step 1: After composing a message and hitting send, your email client - whether it's Outlook Express or Gmail - connects to your domain's SMTP server. This server can be named many things; a standard example would be smtp.example.com.

Step 2: Your email client communicates with the SMTP server, giving it your email address, the recipient's email address, the message body and any attachments.

Step 3: The SMTP server processes the recipient's email address - especially its domain. If the domain name is the same as the sender's, the message is routed directly over to the domain's POP3 or IMAP server - no routing between servers is needed. If the domain is different, though, the SMTP server will have to communicate with the other domain's server.

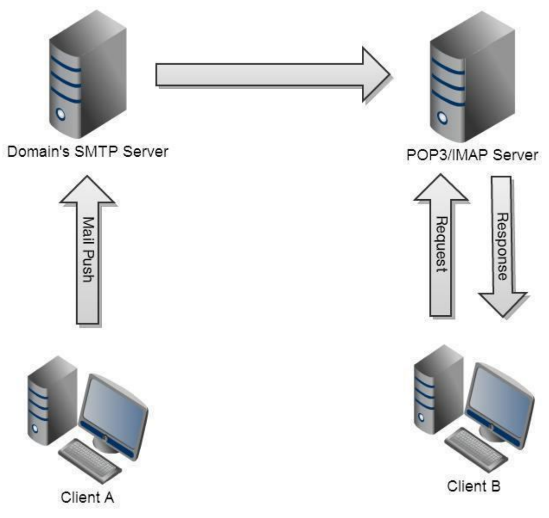


Fig 2: The General Process of sending an email.

Step 4: In order to find the recipient's server, the sender's SMTP server has to communicate with the DNS, or Domain Name Server. The DNS takes the recipient's email domain name and translates it into an IP address. The sender's SMTP server cannot route an email properly with a domain name alone; an IP address is a unique number that is assigned to every computer that is connected to the Internet. By knowing this information, an outgoing mail server can perform its work more efficiently.

Step 5: Now that the SMTP server has the recipient's IP address, it can connect to its SMTP server. This isn't usually done directly, though; instead, the message is routed along a series of unrelated SMTP servers until it arrives at its destination.

Step 6: The recipient's SMTP server scans the incoming message. If it recognizes the domain and the user name, it forwards the message along to the domain's POP3 or IMAP server. From there, it is placed in a sendmail queue until the recipient's email client allows it to be downloaded. At that point, the message can be read by the recipient.

5.1 The Undo Sent Email (USE) Protocol

Now,in the above section we have discussed the actions that happen when we send an e-mail. There are The SMTP is the protocol that transfers the mail from the client to the domain's SMTP server.From there the domains SMTP server identifies whether the recipient's e-mail address is the same as the senders.We will consider this the best case scenario and deal with the worst case scenario later.The SMTP server then transfers the mail to the POP3/IMAP server.

In the POP3/IMAP server we can take ito consideration the fact that each user has a separate space for him in the server.Let us assume these spaces are in the form of folders.Now we will discuss the points of the protocol:

The Best-case Scenario:Sender and Recipient Belong to the same Domain

If the user B(the recipient)has a separate folder for him,then the user A(the sender)will also have a folder for him in the server.(The diagrams below is just to illustrate)

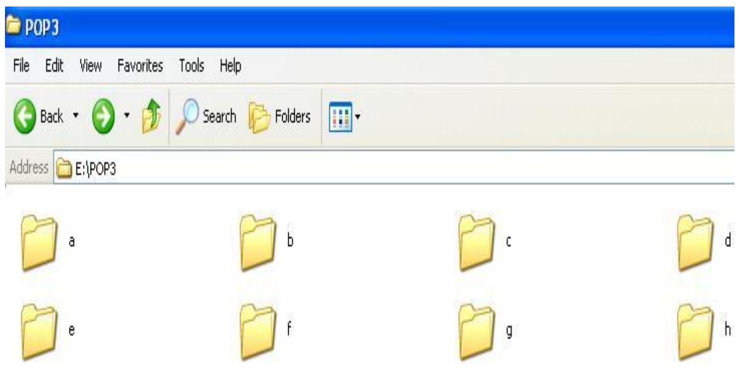
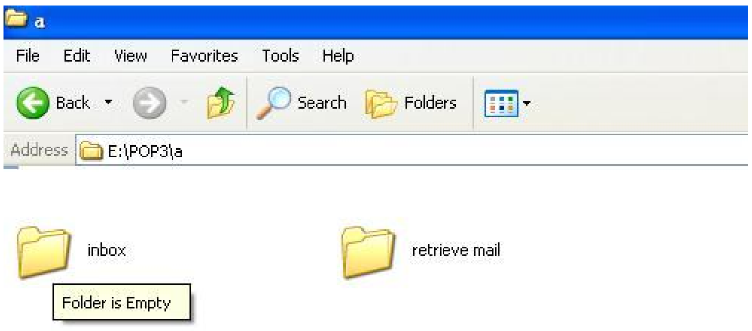


Fig. 3(a): Different users in a POP3 server.



E:\POP3\a

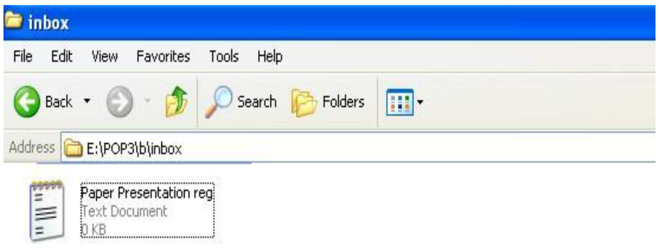
inbox

Folder is Empty

retrieve mail

Fig. 3(b): The added “retrieve mail” folder to all the users.

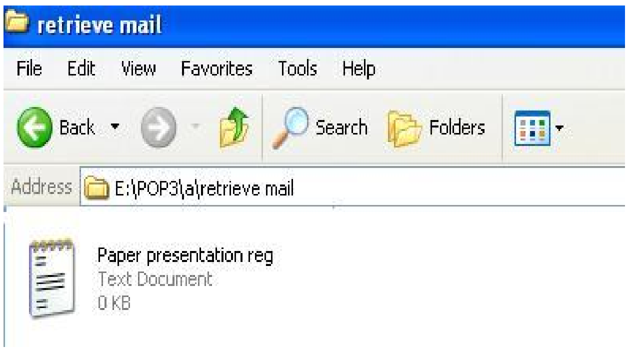
1. In every users folder we add another folder called the—Retrieve Mail||.
2. The domains SMTP server identifies the recipient's(User B)domain and sends it to the —inbox|| folder of the user in the POP3/IMAP server.



E:\POP3\b\inbox

Paper Presentation reg
Text Document
0 KB

Fig. 3(c): The e-mail in the inbox of receiver B.



E:\POP3\a\retrieve mail

Paper presentation reg
Text Document
0 KB

Fig. 3(d): The copy of the sent mail in the retrieve mail of folder of user A

3. We add a extra function to the SMTP server here and send a copy of the mail to the senders(User A) ―Retrieve Mail‖ folder.
4. The POP3/IMAP are both ―pull‖ protocols.
5. The sender who has sent the mail has control over his mail through the retrieve mail folder.
6. If user B has logged into his account then the copy of the mail in the retrieve mail folder gets deleted and hence the user can understand the mail has been read and no longer be read retrieved.
7. If user B has not logged in yet then user A will be able to see the sent mail in the retrieve mail folder and by clicking and downloading the mail the user will be of the imagination that he has retrieved the mail from user B's inbox but actually he has just opened his mail from his own folder.If the user does this the server must delete the mail in the User B's inbox.
8. Thus by these set of actions,the undoing of the sent mail is complete.

The worst-case scenario:Sender and receiver belong to different domains
The worst case scenario for our protocol is when the sender and receiver have accounts in different domains(Eg:User A uses yahoomail and User B uses gmail) because the user may not have an account in the user B's domain.The simplest solution for this is to disable undo in this case but we dont want to deprive the user this option completely.For this purpose we define the set of rules for the worst case scenario:

1. It's not a necessity that a user must specify an alternate e-mail id but it may be asked for security purposes.Our protocol for the worst-case scenario revolves around this.
2. When the sender domain's SMTP server identifies the recipient does not belong to the same domain,it moves the mail to the recpient domain's SMTP server from where it is sent to the domain's POP3/IMAP server.
3. When the sender domain's SMTP server identifies the recipient does not belong to the same domain,it finds out whether the user's altenate e-mail id has the same domain name as that of the recipients and if yes,then the a copy of the mail will be sent to the senders account in the other domain.
4. But the user cannot perform the undo from the account from which he sent the mail.He has to log-in to his alternate email account and perform the undo(provided that email server supports our protocol).

This whole second part of the protocol may look complex but we have to bear in mind we are giving the user the power to undo the sent mail by whatever means possible.But the second part of our protocol does not work when the user has not specified an alternate email id or the user has specified an alternate mail id but the domain name is different from that of the recipients.

5.2 Implementation

A. Program Structure

```
Send()

{

//Originally contains code to send file to the server

//Add extra functionality to copy the mail to the retrieve mail folder in the senders account.

}

Receive()
{

//Originally contains code for the client to get the mail from the server.

//Set a flag to the opened mail.

If(!flag)

{

//Add extra functionality to delete the mail in the retrieve mail folder of the sender.

}

}

Retrieve()

{

//Open the mail in the retrieve mail folder of the sender

//Delete the mail in the inbox of the receiver.}
```

5.3 Advantages and Disadvantages

The USE Protocol will work when:

- Both the sender and receiver belong to the same domain.

  Eg: User A and User B use gmail accounts.

- If both the sender and receiver belong to different domains but the user has specified a alternate email id in which the domain is the same as that of the receiver

Eg: User A has gmail account,User B has yahoomail account and user A has specified that he has an alternate email account in yahoomail.

The USE Protocol wont work when:

- Both sender and receiver belong to different domains and the sender has not specified an alternate email id.
- The sender has specified that he is using an alternate account but the domain is different from that of the receiver.
- If the receiver's domain server doesn't support the USE protocol.

6. Conclusion

The USE protocol provides the end user the ultimate safety net.A wrongly sent mail can threaten the career of an individual(Eg:A wrong mail to your boss) and by this we can undo the wrongly sent mails.We can retrieve the mail that has corrections to be performed and update it and send it again.However it suffers when the domain names of the sender and receiver are different but still the USE protocol can be used as a marketing strategy and this feature of undoing a sent email can draw users towards a particular email service provider.

References

[1] Craig Zacker, ―Networking:The Complete Reference‖
[2] Larry L. Peterson,Bruce S. Davie, ―Computer Networks:A Systems Approach‖, Fourth Edition
[3] Andrew S. Tannenbaum, ―Computer Networks‖, Sixth
[4] Edition,2003,PHI Learning
[5] Kevin Johnson ―Internet Email Protocols: A Developer's Guide‖
[6] www.webopedia.com
[7] www.wikipedia.com
[8] Peer Heinlein (Author), Peer Hartleben (Author) ―The
[9] Book of IMAP: Building a Mail Server with Courier and
[10] Cyrus‖
[11] http://office.microsoft.com/en-us/outlook-help