TRACE e o Correio Multimídia

Ricardo Campanha Carrano, Antonio Caminada,
Débora Christina Muchaluat-Saade, Luiz Claudio Schara Magalhães

Departamento de Engenharia de Telecomunicações - Universidade Federal Fluminense
Rua Passo da Pátria 156, Niterói, RJ

{carrano, antonio, debora, schara}@midiacom.uff.br

Abstract. *Despite its ubiquity, the Internet Electronic Mail System still harbors some pending issues that limit its wide acceptance as an effective multimedia transport platform. In this article we identify some of these issues and show approaches to address them. We demonstrate that the propositions based on the recipient-pull model, in lieu of the current sender-push model, represent the best approaches to address these problems and we highlight the TRACE model being developed at the Universidade Federal Fluminense MidiaCom Lab.*

Resumo. *Apesar da ubiqüidade alcançada, o correio eletrônico da Internet ainda convive com uma série de problemas que limitam seu emprego como plataforma efetiva para o envio de conteúdo multimídia. Neste artigo, identificaremos alguns desses entraves e apresentaremos diversas soluções propostas para solucioná-los. Demonstraremos que as proposições baseadas na reversão do modelo de transporte de sender-push para recipient-pull são um componente fundamental na solução desses problemas e daremos destaque ao modelo TRACE (Transporte de Correio Eletrônico) em desenvolvimento no Laboratório Mídiacom, da Universidade Federal Fluminense.*

1. Introdução

Nas últimas duas décadas, o modelo de transporte de correio eletrônico da Internet, apesar da notável ubiqüidade alcançada e, também, apesar das radicais mudanças em seu perfil de uso, conseguiu permanecer fundamentalmente inalterado. O protocolo SMTP [1] foi ampliado, incorporando novos comandos e funcionalidades [2], mas manteve-se, em seu aspecto mais marcante, o mesmo. Ele ainda se baseia em um sistema de remessa de mensagens do tipo *store-and-forward.*

Nesse artigo apresentaremos os principais obstáculos que precisam ser removidos antes da utilização efetiva do correio eletrônico como plataforma de transporte multimídia/hipermídia e mostraremos como o chamado modelo *pull* pode ser empregado para eliminar ou minimizar estes problemas.

O modelo *recipient-pull*, ou simplesmente modelo *pull*, descreve o cenário em que a informação requerida (um email, no caso) é enviada apenas após a solicitação pelo destinatário, e representa a reversão do atual modelo (*sender-push*) empregado pelo sistema de correio eletrônico em que as mensagens chegam ao seu destino, independentemente do desejo de quem as recebe.

Em nossa análise, mostraremos os problemas que atingem especificamente os conteúdos multimídia (falta de semântica e problemas de armazenamento), e também aqueles que afligem a todas as modalidades de email (o problema do *spam* e a falta de segurança). Mostraremos que, a menos de um problema – o da codificação das mensagens - muitos outros permanecem desafiando a comunidade acadêmica e aguardam por uma solução de ampla aceitação.

Nas seções subseqüentes, apontaremos os desafios existentes no sistema de correio, discutindo os problemas pendentes e os que já foram solucionados. Posteriormente, iremos abordar diversas propostas de melhorias baseadas no chamado modelo *pull*, incluindo o projeto TRACE [3], do Laboratório Mídiacom, da Universidade Federal Fluminense. Finalmente, apresentaremos um comparativo entre essas proposições concorrentes, salientando suas semelhanças e diferenças, assim como procurando apontar as vantagens e desvantagens de cada uma no que diz respeito ao transporte de conteúdos multimídia/hipermídia.

2. Atuais desafios para o Correio Eletrônico na Internet

Nesta seção, apresentaremos o problema da codificação de conteúdos não-textuais em mensagens de correio eletrônico e abordaremos alguns problemas ainda persistentes. Primeiramente, apresentaremos os problemas gerais da falta de segurança e do *spam* e, em seguida, as limitações mais especificamente ligadas ao envio de conteúdo multimídia/hipermídia – o problema da falta de semântica e o problema do ônus de armazenamento das mensagens.

Ao apresentarmos os problemas, apontaremos algumas soluções pontuais que alcançaram graus diversos de aceitação, deixando para a seção subseqüente as propostas que se baseiam na adoção do modelo *pull*.

2.1 O problema resolvido – codificação de conteúdos não textuais

O correio eletrônico foi originalmente desenhado para o transporte de mensagens de texto no idioma inglês, uma tarefa para a qual o conjunto de caracteres ASCII, de sete bits, se prestava satisfatoriamente. Mas esse cenário mudaria com a internacionalização da Internet e com a crescente demanda por envio de arquivos em outros formatos (imagens, vídeos, aúdio e arquivos binários executáveis, dentre outros).

Para acomodar essa demanda, isto é, para permitir o transporte de formatos de arquivos arbitrários através do sistema de correio eletrônico, a solução proposta, e bem sucedida, foi a conversão de formatos.

A idéia é que o conteúdo incompatível original seja convertido para o formato compatível isto é, codificado em sete bits, antes do envio, e decodificado no destino. Foi essa a proposta do MIME (Multipurpose Internet Mail Extensions) [4] que define, para este objetivo, cinco métodos de codificação [5]: 7bit, 8bit, binary, quoted-printable e base64 e, assim, conseguiu resolver uma das demandas relacionadas à utilização do correio eletrônico para o envio de conteúdo multimídia – o transporte propriamente dito. No entanto, como veremos a seguir, outros problemas permanecerm pendentes.

2.2. Falta de segurança

O modelo vigente de transporte de correio eletrônico representa diversas deficiências em relação aos princípios básicos de privacidade e autenticidade.

A privacidade de uma mensagem pode ser comprometida durante seu transporte, já que o protocolo SMTP não prevê o emprego de criptografia. Essa não é propriamente uma deficiência exclusiva do protocolo SMTP e pode mesmo ser enxergada como uma opção de projeto. Afinal, diversas outras aplicações da Internet, lançam mão de implementações de criptografia oferecidas por outros protocolos, das camadas de transporte ou de rede. Um exemplo é o HTTP.

Mas a falta de privacidade de uma mensagem vai além de sua transmissão, alcançando a fase de armazenamento no servidor SMTP de destino. Isso porque mensagens que aguardam leitura estão vulneráveis a usuários com acesso ao servidor.

A solução mais usual tem sido o emprego de criptografia fim-a-fim. Aqui, destacam-se os padrões OpenPGP [6] e S/MIME [7]. Ambos se apóiam na criptografia de chave-pública para garantir a privacidade e também a autenticidade das mensagens.

A questão de autenticidade no correio eletrônico é crítica, sobretudo por sua implicação direta no problema do *spam* onde, como veremos, a verificação de autenticidade não apenas de remetentes, mas também dos próprios servidores de email é vista como uma necessidade.

O uso da criptografia fim-a-fim apresenta a desvantagem de inibir a detecção de software malicioso pelos agentes intermediários (servidores de email com anti-vírus, por exemplo). Mas o maior entrave ao emprego da criptografia no correio eletrônico talvez esteja ligado ao fato de, apesar de incluídas em parte representativa dos MUA (*Mail User Agents*), funcionalidades como o S/MIME ou o OpenPGP não são configuradas por padrão e exigem ações de usuário (criação, distribuição e gerência de chaves) ainda não difundidas suficientemente.

Por isso, uma proposta particularmente interessante é a interposição de um agente intermediário, que possa realizar alterações na mensagem de forma automatizada, isto é, sem a intervenção do usuário ou do agente do usuário. Uma dessas propostas é o GNU Anubis [8] que implementa esta camada entre o MUA e o MTA (*Mail Transfer Agent*), recebendo a mensagem do cliente (MUA), modificando-a para um objetivo específico – como a critpografia, por exemplo – e posteriormente entregando a mensagem alterada ao servidor remetente (MTA).

2.3. *Spam*

No que diz respeito ao correio eletrônico, o combate ao email comercial não solicitado (*spam*) é atualmente o campo de pesquisa que tem recebido maior atenção por parte da comunidade acadêmica. Apesar das diversos modelos propostos para a redução desse tráfego indesejável, o volume de *spam* continua alto e já rompeu a barreira dos 80% das mensagens que circulam pela Internet [9].

A mais difundida das técnicas de combate ao *spam* é o emprego de filtros anti-*spam*, programas que classificam as mensagens recebidas como indesejáveis ou não, em função de uma série de características de seu corpo ou cabeçalho. Esses filtros podem

estar localizados na máquina do usuário ou em seu servidor de email e trazem como principal desvantagem o possibilidade dos falsos positivos – mensagens legítimas que são rotuladas como *spam*.

De adoção também crescente é a técnica de *DNS blacklists*, que consitem de listas mantidas e publicadas na Internet por organizações como DBSL.org [10] ou rfc-ignorant.org [11]. São utilizadas por diversos ISPs, mas apresentam como desvantagem a inclusão nas listas, acidental ou proposital, de sites legítimos.

Outros mecanismos de combate são a cobrança do email [12] e a criminalização do *spam*. Mas muitos defendem a adoção de medidas técnicas mais severas como substituição do protocolo SMTP por um novo protocolo, inteiramente reformulado. Um desses projetos é o IM2000 [13], abordado adiante.

Outras técnicas incluem a autenticação de servidores de email, seja pela recuperação de chaves criptográficas públicas para esses servidores [14] ou através do registro, na base de dados do DNS, dos servidores autorizados ao envio de mensagens de um determinado domínio [15].

E, finalmente, temos a vertente dos que defendem adições ao protocolo SMTP, ao invés de sua substituição. Neste grupo, encontramos propostas como a RFC 3207 [16] que apresenta extensões para permitir a clientes e servidores SMTP o uso de TLS (*Transport Layer Security*), provendo, desta forma, uma comunicação segura e autenticada. Outra proposta nesta linha é o DMTP [17] que introduz novos comandos ao protocolo SMTP. Assim como o IM2000, o DMTP defende o emprego do modelo *pull* no transporte do correio eletrônico (ainda que de forma menos radical) e por isso, voltaremos a essa proposta adiante. De fato, o modelo *recipient-pull* tem sido freqüentemente apontado como uma possível solução para o problema do *spam*.

2.4. A falta de estrutura e de semântica nas mensagens de correio eletrônico

Mensagens de correio eletrônico consistem de corpo e cabeçalho [18]. Graças ao MIME, no entanto, o corpo pode conter uma série de partes arranjadas em uma estrutura de árvore, onde os nós são do tipo *multipart* e contém as folhas (*non multipart*) que correspondem aos conteúdos propriamente ditos.

Exitem vários tipos de nós *multipart* (*alternative, related, signed*, entre outros) e, através deles, podem-se definir inter-relações e hierarquias relativamente complexas. Alguns exemplos usuais de emprego desta hierarquia de conteúdos são:

- Uma mensagem de texto com anexos
- Uma resposta (*reply*) com o original anexado
- O uso de conteúdo alternativo (versão texto e html, por exemplo).

Uma aplicação recente das possibilidades hierárquicas do MIME foi proposta pelo MHTML [19], que define um método para a codificação de documentos HTML em uma mensagem de correio eletrônico. O MHTML introduz um método *inline* para a codificação do documento raiz e todo o conteúdo a que faz referência. Ou seja, em uma mensagem MHTML todo o conteúdo está contido no corpo da mensagem e não referenciado por URLs.

A proposta foi adotada pelo browser Internet Explorer para permitir salvar uma composição HTML em um arquivo único, de formato mht. No entanto, a transmissão de arquivos HTML *inline* não resolve nenhum dos problemas do email multimídia aqui mencionados podendo, inclusive, agravar o problema do tráfego desnecessário, abordado no item a seguir.

Fato é que o MIME, apesar de trazer alguma possibilidade de estrutura às mensagens de correio eletrônico, não introduz a semântica necessária e nem mecanismos que permitam relacionar temporal ou espacialmente suas diversas partes.

Uma proposta para combater a falta de relações semânticas e para a incorporação do conceito de composição em uma mensagem de correio eletrônico é o SMHMS [27] (que também propõe a adoção do modelo *recipient-pull*) e se baseia no chamado Modelo de Contextos Aninhados (Nested Content Model - NCM) [20] para definir composições multimídia. No NCM, um documento é composto por nós (fragmentos de informação) e elos (que definem relações entre nós). Os nós existem em duas classes: terminais e de composição. Os nós terminais contém os dados multimídia propriamente ditos (áudio, vídeo, texto, etc) ao passo que os nós de composição agrupam elos e nós (terminais ou de composição). Os elos conectam nós e são definidos dentro das composições.

É interessante observar que a *World Wide Web* define apenas nós terminais e elos (que pertencem a esses nós terminais), ou seja, não define composições. Essa diferença é relevante na medida em que, como veremos, o modelo da web é encampado pela maior parte das soluções propostas para o correio eletrônico baseadas no modelo *pull*.

No presente artigo, mostraremos que o problema da composição das mensagens é ortogonal aos demais ainda pendentes. Ele trata do *que* está sendo transportado e não de *como* está sendo transportado. A solução para esse problema pode ser a incorporação, por exemplo, de módulos leitores SMIL [29] ou NCL [31] aos MUA.

SMIL é uma linguagem de marcação padronizada pelo W3C [30] que permite a descrição de apresentações multimídia, ou seja, define relações de sincronização, *layout*, animações, transições visuais e conteúdos alternativos, etc. O NCL (Nested Context Language) é uma linguagem de autoria hipermídia, baseada no Modelo de Contextos.

2.5 Otimização de recursos

No modelo de correio eletrônico atual, quando uma mesma mensagem é enviada para um conjunto de n destinatários, são transmitidas e armazenadas n cópias desta mensagem, fato que contribui siginificativamente para o tráfego agregado da rede, principalmente em se tratando de mensagens volumosas (como os conteúdos multimídia, por exemplo).

O protocolo SMTP procura otimizar o envio de mensagens para um mesmo servidor de destino, enviando uma única cópia para os destinatários por este atendidos. No entanto, cópias individuais serão replicadas nas diversas caixas postais locais pelo servidor de destino. Além disso, o número de cópias pode ser ainda maior, por conta de listas de distribuição locais (*aliases* de email).

O problema está diretamente relacionado ao modelo *push* ainda vigente no transporte de correio eletrônico, que transfere o ônus de armazenamento da mensagem, possivelmente desinteressante, para o ambiente do destinatário e, da mesma, forma, inibe a criação de um repositório central para armazenamento da mensagem.

O atual modelo de correio eletrônico pode, portanto, levar à desperdícios de recursos de banda e de espaço de armazenamento. E, o que é pior, penalizando o destinatário em maior escala que o remetente. As proposições *pull*, a seguir abordadas, atacam este problema e também podem auxiliar na minimização dos outros problemas anteriormente listados.

3. Novas propostas para o Correio Eletrônico na Internet

As soluções anteriormente mencionadas atacam pontualmente os problemas do correio eletrônico. O emprego de criptografia, por exemplo, é visto como uma solução natural para as questões de segurança, ao passo que filtros e outras técnicas de coerção procuram combater o *spam*. Mas nenhuma dessas técnicas implica em mudanças reais no modelo transacional do correio eletrônico e, por sua natureza pontual, não dão conta de outros problemas anteriormente identificados.

Motivadas primariamente pelo combate ao *spam*, diversas soluções baseadas no armazenamento de mensagens por conta do remetente têm sido propostas nos últimos anos. Mas apesar desta motivação original, parece razoável supor que, para que alcancem algum sucesso, esses novos paradigmas deverão tratar conjuntamente as deficiências listadas na seção anterior e não apenas o problema do *spam*.

Para efeito de análise, estaremos agrupando as diversas soluções em função do emprego de protocolos que propõem. Enquanto algumas sugerem a extensão do protocolo SMTP, outras defendem sua substituição. Uma terceira categoria, da qual faz parte o modelo TRACE e alternativas similares, propõem a preservação do protocolo SMTP inalterado e o uso conjunto de outros protocolos de aplicação, conforme veremos a seguir.

3.1. Extensão do protocolo SMTP

Desde sua proposição inicial [1], em 1982, o conjunto de comandos do protocolo SMTP foi ampliado em diversas ocasiões, até ser novamente consolidado em 2001 [2]. Nesta consolidação o método de extensão proposto na RFC 1869 [24] foi incorporado e, ainda hoje, serve como base para a adição de novas funcionalidades ao SMTP.

Tabela 1 – Algumas extensões do SMTP

Extensão	Descrição	Definida em
8BITMIME	Permite transmissão de dados codificados em 8 bits	RFC 1652
AUTH	Permite autenticação de cliente e servidor SMTP	RFC 2554
CHUNKING	Melhora a eficiência no envio de arquivos grandes	RFC 3030
PIPELINING	Permite o envio de uma sequência de comandos	RFC 2920
SIZE	Implementa a declaração de tamanho da mensagem	RFC 1870
STARTTLS	Implementa *Transport Layer Security*	RFC 3207

Ao comando de saudação original do SMTP (HELO) foi acrescentado o comando alternativo EHLO. Ao receber o comando EHLO, o servidor deve responder como a lista de extensões que é capaz de suportar. Uma lista de extensões populares é apresentada na Tabela 1 que demonstra que o método de extensões tem sido usado para atacar uma série de problemas do correio eletrônico. Exemplificando, as extensões STARTTLS e AUTH aumentam a segurança do SMTP, ao passo que PIPELINING melhora sua eficiência. Tratando especificamente da eficiência na transmissão de grandes mensagens temos SIZE e CHUNKING, importantes para o envio de conteúdos volumosos, como áudio ou vídeo.

Finalmente, a extensão 8BITMIME dá suporte à codificação em oito bits, vencendo a antiga limitação de codificação em sete bits e aumentando, desta forma, a eficiência no transporte das mensagens. Portanto, podemos afirmar sem grandes riscos, que as extensões são um fator que contribui para a longevidade do SMTP.

Uma das propostas de extensão do SMTP é o Differentiated Mail Transfer Protocol [17]. O DMTP se baseia em uma variação do modelo *receiver-pull*, onde o rementente anuncia sua *intenção* de enviar uma menagem antes do envio propriamente dito. Isso é alcançado através do envio de uma pequena *intention message*. Se houver interesse por parte do destinatário, esse irá contatar o remetente e solicitar a mensagem.

O DMTP é uma técnica de redução do *spam* que se baseia na diferenciação dos remententes, classificados pelo destinatário em "contatos regulares", "*spammers* conhecidos" ou "contatos não classificados" (o estado original de um contato).

Para esse efeito, os autores do DMTP propõem a criação de mais dois comandos para o SMTP – MSID e GTML, usados respectivamente para informar um identificador de mensagem e para recuperar esta mensagem. Temos, portanto, a idéia de uma mensagem de notificação que precede a mensagem propriamente dita. A idéia da mensagem de notificação é, como veremos adiante, capital para a implementação do modelo *pull*.

No entanto, o DMTP não adota o modelo *recipient-pull* de forma integral. Isto porque, segundo a proposta, uma vez que um remetente tenha sido identificado como válido (contato regular) suas mensagens serão recebidas seguindo o modelo usual (*sender-push*). Uma possível justificativa para essa escolha pode vir do fato de que o DMTP é uma solução que visa atacar o problema do *spam*, entendendo que a questão do ônus do armazenamento fica reduzida, uma vez que o *spam* tenha sido eliminado.

A principal desvantagem das soluções baseadas em extensões do SMTP é a necessidade de alteração de servidores e clientes de email ainda que de forma menos extrema do que as propostas apresentadas na próxima seção.

3.2. Substituição do protocolo SMTP

A substituição de um protocolo universal como o SMTP representa um imenso desafio técnico. Ilustrativo desta dificuldade é a lenta adoção da versão 6 do IP. Na Internet, a longevidade dos protocolos aponta para soluções incrementais, em detrimento das substituições.

Dentre as propostas de substituição do SMTP, a mais citada tem sido o IM2000 [13], um projeto que tem como motivação básica a idéia de que o armazenamento das mensagens é responsabilidade do remetente.

Na verdade, esse sistema prevê a substituição não apenas do SMTP, mas também dos protocolos POP e IMAP, por um novo conjunto de protocolos (IMTP, IMRP, IMNP e WYWOP). Um workflow simplificado para o IM2000 seria o seguinte:

1. O remetente *R* compõe a mensagem *M* em seu MUA e a envia. O MUA transfere *M* para o servidor IM2000 local, através do IMTP.
2. O servidor IM2000 marca *M* como nova e envia, através do IMNP, uma mensagem de notificação *N* ao servidor IM2000 do destinatário. *N* inclui um localizador e um ID de *M*.
3. O destinatário *D* acessa seu MUA que se conecta ao servidor IM2000 através do WYWOP, recebendo *N*. Se desejar, *D* poderá recuperar *M* do servidor IM2000 do remetente, através do IMRP.

Cabe acrescentar que nem todas as propostas de substituição do SMTP se baseiam no modelo *pull*. O AMTP [22], por exemplo, é uma proposta de substituição do SMTP cujo foco é a autenticação (Authenticated Mail Transfer Protocol) através do emprego do TLS [23] e de certificados X.509 [24], e se baseia no atual modelo *sender-push*. Já o GIEIS (Global ISP Email Identity System) propõe um sistema mundial de email com verificação centralizada [25] e não a mudança na forma de transporte das mensagens.

Por último, destacamos que nenhuma das propostas desta seção, apesar de já contarem alguns anos (e diversos *ietf drafts*), obteve reverberação significativa. É possível que tenham perdido *momentum* e que prevaleça, pelo menos por mais alguns anos, a idéia de incrementar ao invés de substituir.

3.3. Utilização de modelos híbridos - diversos protocolos

Nos últimos anos, ganharam força muitas propostas baseadas no envio de mensagens de notificação (ao invés da mensagem completa) em um método que, ao mesmo tempo, preservasse a compatibilidade com o SMTP.

A idéia da mensagem de notificação, contendo uma referência para a mensagem original levou, naturalmente, a propostas que incorporavam, de uma forma ou de outra, as técnicas e protocolos da World Wide Web. O conceito de *hyperlink* e de URL foi identificado como uma solução simples e universal para a mensagem de notificação que poderia, evidentemente, ser recuperada através do protocolo HTTP.

Além disso, o HTML acabou sendo adotado como linguagem alternativa para composição das mensagens de email. Uma possibilidade reforçada por esta linguagem ser entendida universalmente, inclusive por *Mail User Agents,* e pela a crescente popularidade do webmail, que tem promovido a integração entre os mundos da web e do correio eletrônico.

Em sua proposta de arquitetura para o email de mídia contínua (CM email), David Turner e Keith Ross [26] descrevem um modelo onde o corpo das mensagens

seria substituído por um *hyperlink* que seria posteriormente utilizado pelo destinatário para recuperar a mensagem original armazenada no servidor do destinatário.

Uma solução similar é o MMM (MultiMedia Mail) [28], uma implementação do SMHMS [27] que utiliza a World Wide Web em vez de NCM como base para criação de mensagens multimídia, o que implica na perda da semântica da composição do SMHMS. Isso visto que, na web, só o que possuímos são elos estáticos de hipertexto (e não referências de sincronização, por exemplo). Por outro lado, o modelo tem como vantagem a utilização de padrões vigentes de ampla aceitação.

De fato, diversas propostas similares parecem validar a idéia de que a desejada reversão para o *recipient-pull* pode se dar através dos protocolos hoje disponíveis, especificamente o HTTP. Neste elenco de soluções se encontra o projeto TRACE, a seguir abordado em maior detalhe.

4. O Projeto Trace

O projeto TRACE [3] é uma proposição do Laboratório Mídiacom da Universidade Federal Fluminense e, como outras soluções aos problemas do correio eletrônico, emprega o modelo *pull* para o transporte do correio eletrônico. Diferente de outras soluções, o TRACE não propõe, para tal, qualquer alteração ou extensão do protocolo SMTP.

Na realidade, como veremos adiante, o modelo de transporte em uso atualmente (Figura 1) é mantido compatível e mesmo os MTAs em uso corrente podem ser utilizados no modelo com a simples adição de um componente TRACE intermediário (TRACE-Proxy), responsável por separar a mensagem em suas partes (cabeçalho e corpo) e, então, utilizar o MTA instalado para envio da mensagem TRACE. Esta, inclusive, é a forma utilizada para testes no Laboratório Midiacom.

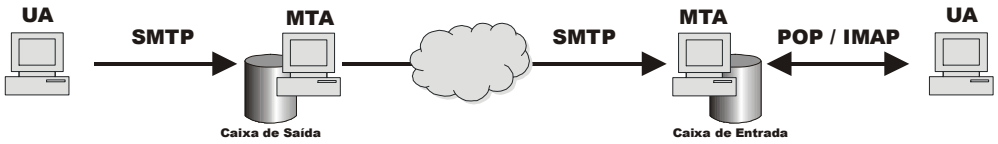

Figura 1: Sistema de Transporte de Correio Eletrônico na Internet

O corpo da mensagem TRACE consiste basicamente de um endereço universal (URL) da mensagem original, a ser recuperada através do protocolo HTTP e tratável, portanto, por praticamente qualquer agente de usuário, o que representa a vantagem da compatibilização automática da base de usuários. O modelo TRACE é, assim, imediatamente compatível, não apenas com os protocolos de transporte e recuperação de email (SMTP, POP e IMAP), como também com os clientes de email (MUA).

No modelo do TRACE (ver Figura 2), quando a mensagem chega ao servidor SMTP do remetente, é feita uma cópia do cabeçalho, adicionando-se mais algumas informações, que permitirão que o destinatário veja a mensagem com segurança, e somente esta cópia é enviada. A mensagem completa fica armazenada. O cabeçalho prossegue via SMTP, como no modelo atual, até chegar ao destinatário, para que este decida então, se deseja baixar a mensagem completa ou não. Caso afirmativo, ele envia uma requisição HTTP ao servidor de armazenamento para que este encaminhe a mensagem.

O TRACE implementa suas funcionalidades através de dois processos: um para dividir a mensagem (separar o cabeçalho do corpo) e enviar o cabeçalho para o destinatário, o TRACE-proxy; e um outro para gerenciar a entrega do corpo, o TRACE-server. Esse último transfere a mensagem quando essa é requisitada, apaga mensagens expiradas e efetua a autenticação de segurança. O *back-end* consiste, assim, desses dois componentes servidores.

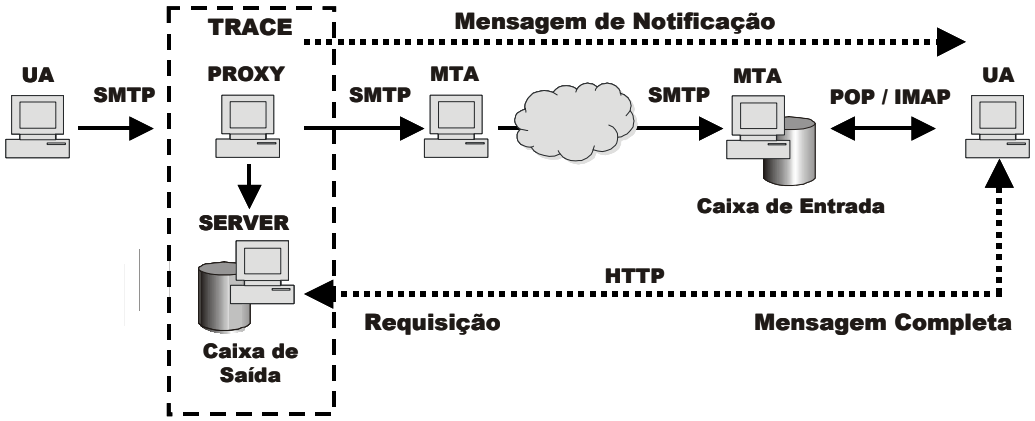

Figura 2: Sistema de Transporte proposto pelo TRACE

O primeiro componente (TRACE-proxy) analisa a mensagem original, separando o cabeçalho do corpo. Ao cabeçalho ele adiciona o ID dessa mensagem, que é usado pelo destinatário para recuperar o corpo da mensagem. O mesmo ID é adicionado ao corpo. Este ID é composto de duas partes: uma que é o identificador de armazenamento, para possibilitar a recuperação da mensagem, e outra, usada para criptografar o corpo da mensagem, conforme será explicado a seguir, para aumentar a privacidade das mensagens armazenadas. Este novo cabeçalho contém, ainda, uma lista dos arquivos anexados (se houver) e um link HTTP para o corpo da mensagem.

O segundo componente (TRACE-server) é responsável pela entrega do corpo da mensagem, quando esse é solicitado pelo destinatário. Esse processo é executado no repositório distribuído, que pode ser uma máquina distinta do MTA do remetente ou não. Ele faz a autenticação do destinatário por intermédio do ID da mensagem, de modo que esse seja o único a ter permissão para ver o seu conteúdo. Para evitar o acúmulo de mensagens, muitas vezes esquecidas pelos destinatários, esse processo também executa um "coletor de lixo", que apaga mensagens após um certo tempo. Ao serem enviadas, as mensagens ganham um "prazo de validade".

Os arquivos anexados são listados no cabeçalho que vai para o destinatário. O corpo da mensagem contém um link para o(s) arquivo(s), para que esses possam ser vistos. Quando é feita a requisição HTTP, eles são enviados e exibidos no browser ou no programa apropriado (ou, então, o usuário pode optar por fazer o download do arquivo), conforme já acontece hoje quando é feita a solicitação de um arquivo a um servidor HTTP.

Para garantir que somente o usuário a quem a mensagem se destina possa lê-la, é usado um esquema de autenticação. Ao ser feita a divisão da mensagem, um identificador é criado e gravado tanto na mensagem quanto no cabeçalho, e este é usado

como chave para a criptografia da mensagem. Ao clicar no link para ver a mensagem, esse identificador é retornado e somente com ele a mensagem pode ser descriptografada para ser lida. Para aumentar a segurança da mensagem, é possível adicionar um esquema de chave privada/chave pública, pois assim nem a captura da mensagem de notificação permitiria a leitura da mensagem, e a mensagem pode ir criptografada até o destino. Se o remetente conhecer a chave pública do destinatário, a mensagem é primeiro criptografada usando essa chave e, depois, a parte do identificador da mensagem é usada para a criptografia simétrica. O identificador é um número de 512 bits escolhido aleatoriamente, de forma a dificultar ataques à base de dados (na qual um atacante tentaria ler mensagens pedindo mensagens em seqüência).

Após ser feita a divisão da mensagem, o cabeçalho (com os campos adicionais) segue o caminho convencional de um correio eletrônico hoje, isto é, segue via SMTP pela rede até o MTA (servidor SMTP) do destinatário e deste para o servidor POP ou IMAP, de onde será transferido pelo usuário para sua máquina. Como o cabeçalho normalmente será bem menor que a mensagem completa, isso permite seu envio a terminais com memória limitada, como telefones celulares e PDAs.

Além dos servidores já desenvolvidos, o projeto TRACE prevê o desenvolvimento de um cliente especializado. Apesar da mensagem TRACE ser compatível com os MUAs atuais, algumas vantagens podem ser alcançadas com um agente de usuário específico. Um exemplo seria uma maior facilidade de incorporar criptografia ao modelo, seja para garantia de sigilo da mensagem ou para verificação de autenticidade do remetente. Outra possibilidade seria a incorporação de capacidades de edição e visualização de documentos SMIL.

O TRACE tem, dessa forma, um efeito colateral desejável: a possibilidade de incorporação prática de soluções de segurança e de técnicas de composição de mensagens, utilizando, para tal, apenas padrões abertos.

Finalmente, a proposta do TRACE é ortogonal a outras propostas de diminuição de *spam*, e pode ser implementada concorrentemente, trazendo no mínimo a diminuição do tráfego na rede, já que mensagens identificadas como *spam* não serão requisitadas e não ocuparão espaço nos servidores de correio dos destinatários. O TRACE também equipara o serviço de email aos outros serviços consagrados na Internet que, utilizando o modelo *pull*, representam um modelo de transporte mais justo, no qual o maior ônus recai sobre o autor.

Para o *proof-of-concept*, um TRACE-proxy foi escrito em java. Este fica entre o MTA original e o MUA, e captura as mensagens, separa cabeçalho e corpo, envia o cabeçalho com os campos adicionais e armazena a mensagem original criptografada. Da mesma forma, um conjunto de scripts php para um servidor Apache implementa o TRACE-server, isto é, recebe pedidos dos destinatários e recupera as mensagens, e também apagar as mensagens expiradas. Este serviço está em operação experimental, mas será usado para oferecer contas de correio eletrônico para os alunos da Escola de Engenharia da UFF, possibilitando um experimento em escala.

5. Comparação entre as propostas

Uma vantagem da proposta TRACE é sua imediata compatibilidade. O TRACE não introduz qualquer mudança nos protocolos de comunicação envolvidos (SMTP e HTTP) como fazem o IM2000 ou, em menor escala, o DMTP e outras soluções baseadas na extensão do SMTP. Como dissemos, substituir ou mesmo alterar um protocolo de ampla aceitação na Internet apresenta grandes dificuldades.

Por outro lado, o TRACE não ataca explicitamente o problema da semântica das mensagens hipermídia como faz o SMHMS. Mas o último é bastante mais complexo em sua arquitetura, de onde se justifica o surgimento do MMM, em muitos aspectos similar ao TRACE e também ao CM email.

Neste ponto é oportuno fazer a distinção entre o método usado para a mensagem de notificação e a linguagem utilizada na composição da mensagem. Parece natural que a mensagem de notificação utilize o conceito de URL e de *hyperlink* para apontar para a mensagem original, mas isso de forma alguma implica em restrições aos métodos de construção da mensagem referenciada.

Assim, tanto TRACE como MMM ou CM email poderiam definir qualquer linguagem de composição para esse efeito. Ou não definir nenhuma, já que o maior entrave a ser removido está ligado ao método de transporte das mensagens, e o problema do formato passa a ser independente do correio eletrônico propriamente dito, uma vez que a mensagem de notificação pode referenciar a qualquer tipo de objeto ou composição.

O princípio de trazer o máximo de melhorias com o mínimo de alterações na infra-estrutura pré-existente reforça a idéia de implementar as mudanças através da interposição de uma camada adicional entre MUA e MTA. Essa é a proposta do GNU Anúbis e também do TRACE.

Em suma, o TRACE ataca primariamente as questões de eficiência e justiça (ônus do armazenamento) e tem como seu principal trunfo a simplicidade e a ortogonalidade a diversas outras propostas de melhoria do modelo de correio eletrônico, particularmente as de combate ao *spam.*

6. Conclusão

As diversas propostas aqui apresentadas, para melhoria do correio eletrônico da Internet, baseiam-se na idéia de que o modelo *sender-push*, hoje vigente, é inadequado e propõem a adoção do paradigma utilizado pela maioria das aplicações cliente-servidor da Internet atual – o modelo *pull*.

Apesar de algumas propostas apresentarem maneiras de incorporar o modelo *pull* através da substituição ou extensão dos protocolos pré-existentes, um caminho mais trilhado tem sido a incorporação de elementos da World Wide Web ao cenário do transporte de correio eletrônico.A maioria destas soluções objetiva atacar primariamente o problema do *spam* e, como efeito colateral, a questão do desperdício de recursos resultante do envio e armazenamento de mensagens replicadas. Mas, é fato que a adoção isolada do paradigma *recipient-pull* não resolve o problema da composição das mensagens multimídia e também não é condição obrigatória para o emprego de criptografia.

Em um primeiro momento, pode parecer que a atual ênfase no combate ao *spam* lançará sombra sobre outros desenvolvimentos importantes no correio eletrônico e pode obstar seu emprego efetivo como plataforma de transporte multimídia/hipermídia. Mas esperamos ter mostrado que o problema do formato e da composição das mensagens é ortogonal aos demais problemas apresentados.

Assim, acreditamos que o emprego de uma linguagem de composição multimídia (como SMIL) aliado à interposição de um Proxy (entre MUA e MTA) que venha a implementar o modelo *pull* e incorpore, ao mesmo tempo, técnicas de criptografia de chave pública, pode representar um enorme avanço para o correio eletrônico. E isso tudo preservando toda a infra-estrutura subjacente atual.

Simultaneamente com a distribuição das contas para alunos da Engenharia da UFF para testes de usabilidade, está sendo desenvolvido um cliente TRACE que permitirá a composição em SMIL e implementará funcionalidades como a criação de uma lista branca para a busca automática de mensagens de remetentes que sejam caracterizados como confiáveis.

Referências

[1] Postel, J. (1982) "RFC 821 - Simple Mail Transfer Protocol"

[2] Klensin, J. (2001) "RFC 2821 - Simple Mail Transfer Protocol"

[3] Caminada, A; Magalhaes, L; (2005) PULL: "Um novo modelo para o Correio Eletrônico." In: XXII Simpósio Brasileiro de Telecomunicações (SBrT'05), 2005

[4] Freed, N; Borenstein N. (1996) "RFC 2045 - Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies"

[5] Freed, N; Klensin, J. (2005) "RFC 4289 - Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures"

[6] OpenPGP - IETF OpenPGP working group - http://www.ietf.org/html.charters/openpgp-charter.html (acessado em 20/12/2006)

[7] S/MIME - S/MIME Mail Security - http://www.ietf.org/html.charters/smime-charter.html (acessado em 20/12/2006)

[8] Anubis – GNU Anubis - Free Software Foundation (FSF) - http://www.gnu.org/software/anubis/ (acessado em 20/12/2006)

[9] Messaging Anti-Abuse Working Group - Email Metrics Program: The Network Operators' Perspective - Report #2 - 1st Quarter 2006 - http://www.maawg.org/about/FINAL_1Q2006_Metrics_Report.pdf

[10] Distributed Sender Blackhole List - http://dsbl.org/main. (acessado em 20/12/2006)

[11] RFC-Ignorant.Org - http://www.rfc-ignorant.org (acessado em 20/12/2006)

[12] E-Postage - Method for Controlling Spam Via E-Postage Fees - http://www.mall-net.com/spam/ (acessado em 20/12/2006)

[13] IM2000 - Electronic mail transport of the future - http://www.im2000.org/ (acessado em 20/12/2006)

[14] IIJ Introduces Sender Authentication Technology - http://www.ip97.com/iij_introduces_sender_authentication_fdc.aspx

[15] SPF - Sender Frame Policy – www.spf.org (acessado em 20/12/2006)

[16] Hoffman, P. (2002) “RFC 3207 - SMTP Service Extension for Secure SMTP over Transport Layer Security”

[17] Duan, z.; Gopalan, k; Dong, Y. (2005) "DMTP: Controlling Spam Through Message Delivery Differentiation"

[18] Resnick P. (2001) “RFC 2822 - Internet Message Format”

[19] Palme, J; Hopmann, A; Shelness, N. (1999). “RFC 2557 - MIME Encapsulation of Aggregate Documents, such as HTML (MHTML)”

[20] MCA (2003) - Modelo de Contextos Aninhados (versão 2.2) - Relatório Técnico FT-TM-SH-000 - Laboratório telemídia - Departamento de informática - Puc-rio

[21] Klensin, J; Freed, N; Rose, M; Stefferud, E; Crocker, D. (1995) “RFC 1869 - SMTP Service Extensions”

[22] AMTP – Authenticated Mail Transfer Protocol - http://amtp.bw.org/docs/ (acessado em 20/12/2006)

[23] Dierks, T; Allen, C. (1999) “RFC 2246 - The TLS Protocol”

[24] Housley, R; Polk, W; Ford, W; Solo, D. (2002) “RFC 3280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile”

[25] Global ISP Email Identity System- http://pribadi.or.id/diary/2003/07/10/gieis-global-isp-email-identity-system/ (acessado em 20/12/2006)

[26] Turner, David A.; Ross, Keith W. (2000) "A Comprehensive Architecture for Continuous Media E-mail".

[27] SMHMS: Um Correio Eletrônico Multimídia/Hipermídia - ftp://ftp.telemidia.puc-rio.br/pub/docs/conferencepapers/1994_05_SOARES.pdf (acessado em 20/12/2006)

[28] Batista, T.; Rodriguez, N. de La Rocque; Soares, L. F. Gomes, Resende, M.C. (1996) "MMM: Um Correio Eletrônico Multimídia sobre o WWW"

[29] SMIL - The Synchronized Multimedia Integration Language - http://www.w3.org/AudioVideo/

[30] W3C - The World Wide Web Consortium – http://www.w3c.org

[31] Muchaluat-Saade, D. C; Silva, H. V. de Oliveira; Soares, L. F. Gomes. (2003) Linguagem NCL versão 2.0 para Autoria Declarativa de Documentos Hipermídia. IN: IX Simpósio Brasileiro de Sistemas Multimídia e WEB - WebMídia 2003 - WebMídia2003