Filtr protokołu SMTP

Praca inżynierska

Zbigniew Artemiuk
Z.Artemiuk@stud.elka.pw.edu.pl

23 lipca 2008

Spis treści

1 Wstęp

Obecnie w Internecie, pomimo szeregu rozwijających się form komunikacji, chociażby technologii takich jak tekstowa komunikacja czasu rzeczywistego (czyli IM - instant messaging i popularne chaty) czy też jeszcze bardziej zaawansowanych technologii, które umożliwiają przesyłanie głosu oraz wideo (chociażby Skype), pospolite maile cały czas znajdują zastosowanie. Używamy ich chyba obecnie najczęściej do kontaktów biznesowych, rodzinnych, przyjacielskich, głównie w celach informacyjnych (czyli sam tekst), jak również do przekazywania niewielkich plików. Chyba każdy internauta legitymuje się przynajmniej jedną skrzynką mailową (a czasami jest ich znacznie więcej). Każdy spotkał się także z niechcianą pocztą (niechcianymi mailami), nazywaną bardzo ogólnym określeniem Spam, i narzędziami (chociażby tymi wbudowanymi w klientów pocztowych dostępnych z poziomu przeglądarki internetowej), które umożliwiają odseparowanie takiej poczty, od tej która niesie interesującą nas zawartość merytoryczną. Separacja ta to tzw. filtrowanie ze względu na obiekt filtracji zwane filtracja poczty. Dokonanie filtracji wymaga jednak poznania dokładnie w jaki sposób transportowane są nasze maile, a to wszystko zawarte jest w protokole Simple Mail Transfer Protocol (w skrócie SMTP). Coś tutaj jeszcze dopisać o protokole, o jego wadach, o konieczności filtracji.

2 Cel i zakres pracy

Celem pracy jest zaprojektowanie oprogramowania, które po uruchomieniu umożliwiałoby filtrację wiadomości w ramach protokołu SMTP. Oprogramowanie to byłoby konfigurowalne poprzez plik tekstowy zawierający reguły dotyczące wiadomości SMTP i zachowania oprogramowania po napotkaniu takiej wiadomości. Reguły te dawałyby możliwość następującej filtracji:

- wykrywanie wiadomości posiadających bądź nie posiadających określony nagłówek (nagłówki)
- wykrywanie wiadomości posiadających bądź nie posiadających określonej frazy w nagłówku (nagłówkach)
- wykrywanie wiadomości posiadających bądź nie posiadających określonej frazy w nagłówkach części maila (gdy nie jest to prosty mail)
- wykrywanie określonych content-type-ow w wiadomościach
- wykrywanie wiadomości o określonej wielkości części (jeżeli wiadomość jest jednoczęściowa to tyczy się treści wiadomości)

Po natrafieniu na wiadomość spełniającą kryteria danej reguły zostanie ona w zależności od zapisu w konfiguracji:

- odrzucona (wyfiltrowana)
- przepuszczona, a jej obecność zostanie zaznaczona w logach
- przepuszczona

3 Protokół SMTP

3.1 Historia powstania

3.1.1 ARPANET czyli początki Internetu

Historia powstania protokołu SMTP jest ściśle związana z początkami Internetu. Internet zaś i jego kreowanie związane jest bezpośrednio ze swoim przodkiem czyli ARPANETEm.

ARPANET (Advanced Research Projects Agency Network) został stworzony przez przez jedną z agencji United States Department of Defense (departament bezpieczeństwa Stanów Zdjednoczonych) o nazwie ARPA (Advanced Research Projects Agency). Nazwa agencji została później przekształcona na DARPA (D od Defence). Agencja ta miała zając się rozwojem nowych technologii na potrzebę amerykańskiego wojska.

W miarę wchodzenia w życie komputerów wykorzystywanych w ramach agencji powstała idea stworzenia sieci pomiędzy nimi, która to umożliwiłaby komunikacje pomiędzy ich użytkownikami. Idea ta została po raz pierwszy zaproponowana przez Josepha Carla Robnetta Licklidera z firmy Bolt, Beranek and Newman (obecnie BBN Technologies) w sierpniu 1962 w serii notatek na temat koncepcji "Międzygalaktycznej Sieci Komputerowej". Zawierała ona prawie wszystko czego możemy doświadczyć w dzisiejszym Internecie.

W październiku 1963 roku Licklider został mianowany szefem programu Behavioral Sciences and Command and Control w ARPA. Przekonał on wtedy Ivana Sutherlanda i Boba Taylora, że jego wizja jest czymś naprawdę istotnym. Sam Licklider nie doczekał jednak żadnych konkretnych prac w kierunku jej urzeczywistnienia, gdyż opuścił ARPA.

ARPA i Taylor cały czas byli zainteresowanie stworzeniem sieci komputerowej, ażeby zapewnić naukowcom pracującym w ramach ARPA w różnych lokalizacjach, dostęp do innych komputerów, które firma oferowała. Istotne było także, aby nowe oprogramowanie i rezultaty badań były jak najszybciej widoczne dla każdego użytkownika sieci. Sam Taylor posiadał 3 różne terminale, które dawały mu podłączenie do 3 różnych komputerów - jeden do SDC Q-32 w Santa Monica, drugi w ramach projektu Project Genie do komputera na Uniwersytecie w Kalifornii (Berkley) i ostatni do komputera z Multicsem w MIT (The Massachusetts Institute of Technology).

Taylor w taki sposób opowiadał od podłączeniu do tych komputerów: " Dla każdego z tych terminali miałem inny zestaw poleceń. Dlatego też kiedy rozmawiałem z kimś z Santa Monica, a poźniej chciałem ten sam temat skonsultować z kimś z Berkley albo MIT, musiałem przesiąść się do innego terminala. Oczywistym wtedy wydało mi się, że musi być 1 terminal, który obsłuży te 3 połączenia. Idea ta to właśnie ARPANET"

Do połowy 1968 roku kompletny plan sieci został stworzony i po zatwierdzeniu przez ARPA, zapytanie ofertowe RFQ (Request For Quotation) zostało posłane do 140 potencjalnych wykonawców. Większość potraktowała propozycję jako dziwaczną. Tylko 12 firm złożyło oferty z czego 4 zostały uznane za najważniejsze. Do końca roku wyłoniono 2 firmy, z których ostatecznie 7 kwietnia 1969 roku została wybrana firma BBN.

Propozycja BBN była najbliższa planom ARPA. Pomysłem ich było stworzenie sieci z mały komputerów zwanych Interface Message Processors (bardziej

znanych jako IMPs), które to obecnie nazywamy routerami. IMPsy z każdej strony zapewniały funkcje przechowywania i przekazywania pakietów, a połączone były między sobą przy użyciu modemów podpiętych do łączy dzierżawionych (o przepustowości 50 kbit/sekundę). Komputery podłączone były do IMPsów poprzez specjalny bitowy interfejs. W ten sposób stawały się one częścią sieci ARPANET.

Do zbudowania pierwszej generacji IMPsów BBN wykorzystała komputer Honeywell DDP-516. Został on wyposażony w 24kB pamięci rdzenia (z możliwością rozszerzenia) oraz 16 kanałów Direct Multiplex Control (DMC) do bezpośredniogo dostępu do tej pamięci. Poprzez DMC podłączane były komputery użytkowników (hosty) i modemy. Dodatkowo 516 otrzymał ezstaw 24 lamp, które pokazywały status kanałów komunikacyjnych IMPa. Do każdego IMPa można było podłączyć do czterech hostów i mógł się on komunikować z 6 zdalnymi IMPami poprzez współdzielone łącza.

Zespół z BBN (początkowo 7 osób) szybko stworzył pierwsze działające jednoski (IMPy). Cały system, który zawierał zarówno sprzęt jak i pierwsze oprogramowania zarządzające pakietami, został zaprojektowany i zainstalowany w ciągu 9 miesięcy.

Początkowo ARPANET składał się z 4 IMPów. Zostały one zainstalowane w:

- UCLA (University of California, Los Angeles), gdzie Leonard Kleinrock założył centrum pomiaru sieci (Network Measurment Center)
- The Stanford Research Institute's Augmentation Research Center, gdzie Douglas Engelbert stworzył system NLS, który między innymi wprowadził pojęcie hypertextu
- UC Santa Barbara
- The University of Utah's Graphics Department, gdzie przebywał ówcześnie Ivan Sutherland

3.1.2 Piersze wiadomości w ARPANETcie

Pierwsza komunikacja host-host w sieci ARPANET wykorzystywała protokół 1822, który definiował sposób w jakis host przesyłał wiadomość do IMPa. Format wiadomości był tak zaprojektowany, żeby bez problemu mógł pracować z szerokim zakresem architektur. Zasadniczo wiadomość składała się z:

- typu wiadomości
- adresu hosta
- pola z danymi

W celu wysłania wiadomości do innego hosta, host wysyłający powinien sformatować wiadomość tak, aby ta zawierała adres hosta docelowego oraz dane, a następnie dokonać transmisji wiadomości przez interfejs sprzętowy 1822. IMP dostrzeże dostarczenie wiadomości albo poprzez dostarczenie jej bezpośrednio do hosta docelowego albo poprzez przekazanie jej do kolejnego IMPa. Kiedy wiadomość została odebrana przez docelowego hosta, IMP do którego host był

podłączony wysyła potwierdzenie odbioru (zwane Ready for Next Message or RFNM) do hosta wysyłającego.

W przeciwieństwie do obecnych protokołów datagramowych w Interecie (takich jak no IP), ARPANETowy porotokół 1822 zapewniał niezawodność w taki sposób, że informował o niedostarczonej wiadomości. Niemniej protokół 1822 nie był odpowiedni do żonglowania wieloma połączeniami w różnych aplikacjach uruchomionych na pojedynczym hoście. Problem ten został rozwiązany dzięki wprowadzeniu na hostach Network Control Program (NCP), dzięki któremu możliwe było niezawodne, z kontrołą przepływu, dwukierunkowe połączenia pomiędzy różnymi procesami na różnych hostach. NCP implementował kolejną warstwę znajdującą się na górze stosu protokołów. Dzięki niemu aplikacje, które miały mieć już jakąś konkretną funkcjonalność, mogły wykorzystywać spójny interfejs i korzystać swobodnie z dobrodziejstw ARPANETu czyli wykonywać połączenia do innych aplikacji przez sieć.

3.1.3 SNDMSG i READMAIL

Na początku roku 1970, powstał program (a w zasadzie 2 oddzielne) do wysyłania i odbierania wiadomości. Zaimplementował go Ray Tomlinson. Programy te to SNDMSG i READMAIL. Pierwsza wersja tych programów, była kolejną implementacją, która umożliwiała wymiane informacji między użytkownikami jednej maszyny (jednego hosta), a konkretnie komponowanie, adresowanie i wysyłanie wiadomości do skrzynek użytkowników. Już jednak w 1971 Tomlinson stworzył pierwszą aplikację ARPANETową (w ramach prac nad systemem TENEX), która umożliwiała wysyłanie wiadomości do dowolnych hostów. Tomlinson dokonał usprawnień w programie SNDMSG przy okazji pracy nad projektem CPYNET, którego celem było stworzenie protokołu do wymiany plików między komputerami w sieci.

Skrzynkę emailową był wówczas był zwykły plik o określonej nazwie. Specjalną właściwością jego było to, że miał ochronę taką, że umożliwiał innym użytkownikom dopisywanie do pliku. Mogli oni więc pozostawiać kolejne wiadomości ale nie mogli ich czytać ani nadpisywać wcześniejszych (jedynie właściciel mógł to robić). Tomilson zauważył, że CPYNET może dopisywać zawartość do pliku skrzynki mailowej, na takiej zasadzie jak SNDMSG (SNDMSG umiał zapisywać wtedy tylko na lokalnej maszynie). Dlatego też SNDMSG mógł po prostu skorzystać z kodu CPYNET i bezpośrednio dostarczać wiadomości poprzez połączenie sieciowe do zdalnych skrzynek poprzez dopisywanie kolejnych informacji do plików na innych hostach.

Brakującą częścią była możliwość dopisywania do plików przy pomocy CPYNET. Dotychczas mógł on tylko słać i odbierać pliki. Dodanie tej funkcjonalności nie było czymś wielkim dla twórców protokołu i funkcjonalność ta wkrótce zainstniała.

Następnie Tomilson włączył kod CPYNET do SNDMSG. Pozostało jedynie rozróżnienie maili lokalnych od maili zdalnych. Dlatego też Tomilson zdecydował się, że maile zdalne będą rozpoznawane po tym, że po loginie (czyli wyznaczniku użytkownika do którego wiadomość jest słana) nastąpi specjalny znaczek @ (ang. at, a polska znana wszystkim małpa) , a tuż po nim nazwa hosta czyli zdalnego komputera, na którym skrzynkę ma dany loginem użytkownik. Tomilson tak powiedział o wyborze małpy jako znaczka rozdzielającego login od hosta: "I am frequently asked why I chose the at sign, but the at sign just makes

sense" (w wolnym tłumaczeniu: Jestem często pytany czemu wybrałem znaczek małpy, ale on po prostu miał sens).

Pierwsza wiadomość została przesłana pomiędzy maszynami, które fizycznie były obok siebie. Jedyne połączenie jakie było między maszynami (oprócz podłogi ;]) było za pośrednictwem sieci ARPANET. Tomilson przesłał wiele wiadomości do samego siebie z jednej maszyny na drugą. Pierwsze treści wiadomości od razu zostały zapomniane. Najprawdopodobniejsza ich treść była w stylu QWERTYUIOP. Kiedy Tomilson był zadowolony z programu wysłał do swoich kolegów z zespołu wiadomość z instrukcją jak słać wiadomości przez sieć. I tak pierwsza wysłana wiadomość ogłosiła swoje istnienie.

Te pierwsze wiadomości zostały wysłane pod koniec 1971 roku. Następne wydanie TENEXa, które ukazało się w 1972 roku, zawierało SNDMSG, z możliwością wysyłania maili przy użyciu sieci ARPANET. Protokół CPYNET wkrótce został zastąpiony prawdziwym protokołem do transferu plików i posiadał specyficzne dodatki do obsługi maila.

3.1.4 MAIL i MLFL w protokole FTP

Protokołem, o którym wcześniej była mowa był protokół FTP, którego specyfikacja wstępna została zawarta w RFC (Request For Comment) 114. Kolejne ulepszenia protokołu zostały dokonana w roku 1972 i wtedy też weszły do niego nowe polecenia pozwalające korzystać ze skrzynek emailowych. Poleceniami tymi były:

- MAIL - polecenie to pozwalało użytkownikowi przy pomocy telnetu wysłać maila. Pomocne to było gdy użytkownik ten nie korzystał ze swojego hosta i logował sie do niego poprzez protokól telnet
- MLFL - polecenie to pozwalało na normalne skonstruowanie maila i przesłanie odpowiednio wskazanego pliku jako jego treści

Protokół ten stał się aż do roku 1980 standardem przesyłania emaili w ARPANETcie. Wtedy to został wypart przez protokół SMTP, który z pewnymi usprawnieniami funkcjonuje aż do dziś.

3.1.5 Rozwój innych programów do obsługi poczty

Zanim jednak powstał protokół SMTP cały czas pracowano nad rozwojem ówczesnych programów chociażby do odbioru maili. W ten sposób na prośbe Steve'a Lukasika Lawrence Roberts, ówczesny dyrektor IPTO, stworzył program RD, który składał się z makr w edytorze TECO (Text Editor and COrrector).

Program RD umożliwiał:

- sortowanie emaili po temacie i dacie
- czytanie, zapisywanie i czytanie wiadomości w dowolnym porządku

RD nie powstał więc jako wynik badań, ale z czysto praktycznej potrzeby swobodnego zarządzania emailami.

Wkrótce powstały też kolejne ulepszenia programu RD oraz SENDMSG, takie jak NRD czy WRD, które to wprowadzały kolejne ulepszenia istniejących już implementacji.

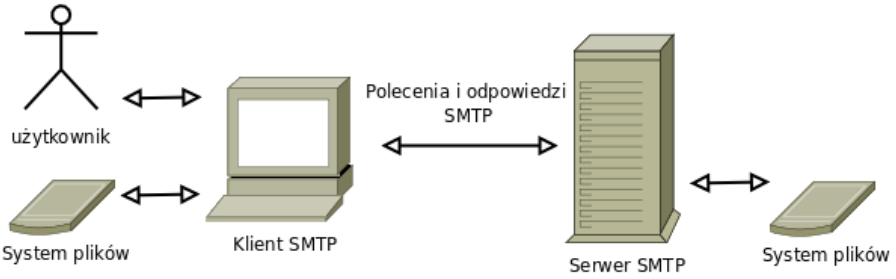
Warte wspomnienia jest także powstanie programu MSG, który umożliwiał między innymi przekazywanie maili (ang. forwarding), konfigurowalny interfejs, czy też polecenie Odpowiedz (ang. Answer), które automatycznie uzupełniało adres odbiorcy. MSG można było nazwać pierwszym nowoczesnym programem pocztowym.

W 1977 roku różne formaty wiadomości zostały zebrane w jedną spójną specyfikację co doprowadziło do stworzenia RFC 733. Specyfikacja ta połączyła wcześniej istniejące dokumentację z odrobioną innowacji i była jednocześnie pierwszym RFC, które deklarowało standard Internetowy (ówcześnie ARPANETowy).

3.2 Szczegóły protokołu

W 1982 roku powstało główne RFC opisujące protokół SMTP - RFC 821. W 2001 roku zostało ono rozbudowane i powstał dokument, który jest obowiązującą obecnie specyfikacją protokołu. Informacje te zawarte zostały w RFC o numerze 2821.

3.2.1 Model protokołu, podstawy



Komunikacja w protokole SMTP oparta jest na nastepującym modelu: kiedy klient SMTP posiada wiadomość, którą chce przetransmitować ustanawia dwukierunkowy kanał transmisyjny z serwerem SMTP. Klient SMTP bierze na siebie odpowiedzialność za przetransportowanie wiadomości do jednego lub wielu serwerów SMTP, lub zgłoszenie błędu jeśli operacji przekazania maila nie powiodła się.

Adres serwera SMTP klient determinuje poprzez zamianę domeny podanej w adresie na pośredni host (Mail eXchanger) lub ostateczny host docelowy.

Serwer SMTP może być ostatecznym celem albo też pośrednim "przekaźnikiem" (po odebraniu wiadomości to on może przejąć rolę klienta SMTP) albo "bramą" (może transportować wiadomość dalej używając innego protokołu niż SMTP). Polecenia protokołu SMTP generowane są przez klienta i posyłane do serwera. Odpowiedzi słane są jak reakcja na żądania klienta.

Innymi słowy wiadomość może być przetransportowana w trakcie pojedynczego połączenia pomiędzy pierwotnym adresatem, a ostatecznym klientem albo może składać się z kilku pośrednich połączeń. W obu przypadkach zachowana jest odpowiedzialność za wiadomość - protokół wymaga od serwera wzięcia odpowiedzialności za dostarczenie wiadomości albo w przypadku nieprawidłowości za zgłoszenie błędu.

Kiedy kanał transmisyjny zostanie zestawiony klient SMTP inicjuje transakcję maila. Składa się ona z serii poleceń, które mają na celu wyspecyfikowanie

nadającego i celu wiadomości, a następnie przesłania jej zawartości (oczywiście z nagłówkami, które wchodzą w jej skład). Kiedy ta sama wiadomość jest adresowana do wielu odbiorców, protokół przesyła jedną kopię wiadomość dla wszystkich odbiorców w obrębie jednego hosta.

Serwer reaguję na każdą komendę odpowiedziami. Wskazują one akceptację komendy (tej przesłanej od klienta) i oczekiwanie na kolejne, bądź też wystąpienie tymczasowych albo stałych błędów.

Kiedy mail zostanie przetransmitowany, klient może zamknąć połączniee albo też zainicjować kolejną tranzakcję innego maila. Dodatkowo klienty SMTP mogą używać połączenia z serwerem w celach np. weryfikacji adresu email bądź też uzyskanie adresów subskrybentów listy dyskusyjnej.

Jak zostało wcześniej zasugerowane transmisja w protokole może odbywać się bezpośrednio pomiędzy hostem ślącym wiadomośc, a hostem docelowym, kiedy są one połączone tym samym serwisem transportowym. Kiedy tak nie jest, transmisja odbywa się poprzez hosty pośrednie. Hosty te wybierane są poprzez mechanizm serwera domen (DNS) zwany Mail eXchanger.

3.2.2 Model rozszerzeń

W wyniku prac, które rozpoczęły się około dekadę po wydaniu pierwszego RFC dotyczącego protokołu SMTP (czyli RFC 822), protokół został rozszerzony i pojawiły się w nim nowe funckjonalności. Standardowy model został więc zmodyfikowany o dodatkowe serwisy. Pozwalają one na ustalenie pomiędzy klientem, a serwerem usług dodatkowych jakie są oni w stanie obsłużyć (poza podstawowymi wymaganiami SMTP). Mechanizm rozszerzeń SMTP określa środki, za pomocą których klient i serwer mogą dokonać wzajemnego ropoznania, a serwer może także poinformować klienta o rozszerzeniach, które on wspiera.

Współczesna implementacja SMTP musi obsługiwać podstawowe mechanizmy rozszerzeń. Przykładowo serwer musi wspierać komendę EHLO, jeśli nawet nie implementuje innych specyficznych rozszerzeń, natomiast klient powinien skłaniać się ku używaniu komendy EHLO niż HELO.

SMTP jest szeroko używany i pojawiło się wiele bardzo dobrych implementacji protokołu wyposażonych w liczne rozszerzenia. Społeczność w Internecie jednak uważa, że niektóre z serwisów są bardzo ważne, mimo tego, że nie powstały gdy protokół był po raz pierwszy projektowany. Jeśli serwisy te mają zostać dodane to w zachowaniem zgodności wstecz.

Obecnie szkielet rozszerzeń składa się z:

- Polecenia EHLO, które powinno być używane zamiast wcześniejszego HELO
- Rejestr rozszerzeń SMTP
- Dodatkowe parametry do poleceń SMTP MAIL i RCPT
- Opcjonalnych zastąpień poleceń w protokole SMTP, jak na przykład DATA w transmisji nie przy pomocy ASCII (ang. pipelining opisany później)

Siłą SMTP jest jego prostota. Doświadczenia z protokołami pokazują, że te z kilkoma możliwościami stają się powszechnie używanymi, podczas gdy te z wieloma możliwościami przestają być czytelne.

3.2.3 Sesja protokołu

Sesja jest inicjowana kiedy klient otwiera połączenie do serwera i serwer odpowie wiadomością zapraszającą.

SMTP serwer po podłączeniu się klienta może po kodzie powitalnym 220 poinformować klienta o swoim oprogramowaniu i jego wersji.

220 mail.foo.com SuperSMTP v 6.1.2 Service ready

Protokół SMTP pozwala serwerowi na odmówienie transakcji podczas gdy formalnie umożliwia na podłączenie się do serwera SMTP. Wtedy zamiast kodu 220 serwer odpowiada kodem 554.

554 Transaction failed (Or, in the case of a connection-opening response, "No SMTP service here")

Serwer w tym przypadku musi czekać zanim zakończeny połączenie na komende QUIT ze strony klienta. W trakcie tego oczekiwania wszystkie inne komendy zostaną potraktowane odpowiedzią 503.

503 Bad sequence of commands

Kiedy serwer przywita się z klientem i klient otrzyma wiadomość powitalną, standardowo wysyła on polecenie EHLO do serwera i w wiadomości tej przedstawia się.

EHLO bar.com

Użycie komendy EHLO wskazuje, że klient jest w stanie obsłużyć rozszerzenia protokołu i prosi serwer o ich listę. Starsze systemy SMTP, które nie obsługują rozszerzeń protokołu czy też starsze klienty SMTP, które nie wymagają tychże rozszerzeń, mogą używać komendy HELO.

HELO bar.com

Serwer w odpowiedzi na HELO nie powinien zwrócić listy rozszerzeń. Jeżeli serwer zwróci na EHLO

500 Command not recognized

, wtedy klient powinien być w stanie przesłać komende HELO.

Następnie dochodzi do transakcji maila. Rozpoczyna się ona poleceniem MAIL, która identyfikuje nadawcę. Po niej następuje seria poleceń RCPT, które przekazują informację o odbiorcy, a na sam koniec DATA czyli ciało maila, które kończone jest wyznacznikiem koća maila. Znacznik ten potwierdza transakcję.

Pierwszym krokiem w procedurze transakcji jest komenda MAIL. Ogólna jej składnia wygląda następująco:

MAIL FROM:<reverse-path> [SP <mail-parameters>] <CRLF>

przykładowo

MAIL FROM:<bob@example.org>

Poleceni to mówi odbiorcy SMTP (serwerowi), że rozpoczynana jest nowa transakcja. Dlatego powinien on zresetować (chyba na danym połączeniu?) wszystkie poprzednie stany tabel i buforów, włączając w to odbiorców i dane maila. Argument <reverse-path> zawiera adres źródłowej skrzynki mailowej, która może posłużyć w celu informacji o ewentualnych błędach (będzie dalej opisane [4.2]). Jeśli polecenie zostanie zaakcpetowane serwer odpowiada poleceniem 250 OK.

250 OK

Jeśli przesłany adres nie został zaakceptowany serwer musi odpowiedzieć czy błąd jest stały (zawsze będzie się pojawiał przy tym adresie) czy tymczasowy

(klient będzie mógł wykonać ponowną próbę, która może zakończyć się sukcesem). Czasami jednak serwer może dokonać akceptacji adresu przekazanego w poleceniu MAIL i dopiero po analizie adresów przesłanych w poleceniu RCPT, może zgłosić błąd. Normalnie jednak gdy błąd jest stały odpowiedź serwera to

550 mailbox not found

albo tymczasowy

503 i co tutaj?

Historycznie <reverse-path> może nie tylko zawierać adres skrzynki, jednakże obecne oprogramowanie nie powinno używać routingu źródła.

Parametry opcjonalne w poleceniu MAIL <mail-parameters> służą do negocjacji rozszerzeń protokołu i będą opisane później. (2.2 sekcja - to juz pisalem - moze gdzies dalej będzie napisane co dokładnie mogą tutaj gadać)

Następnym krokiem w transakcji jest polecenie RCPT. Jego składnia to

RCPT TO:<forward-path> [SP <rcpt-parameters>] <CRLF>

Argumentem tego polecenia jest adres (z reguły adres skrzynki oraz domena) identyfikujący odbiorcę. Jeśli zostanie on zaakceptowany przez serwer odpowiada on komunikatem 250 OK i zapisuje sobie argument forward-path. Jeżeli adres nie jest prawidłowy serwer odpowiada kodem 550. Z reguły odpowiedź wygląda tak

550 No such user

Procedura ta może być powtórzona parokrotnie przykładowo

C: MAIL FROM:<bob@example.org> S: 250 Ok C: RCPT TO:<alice@example.com> S: 250 Ok

W tym przypadku kopia maila trafi do 2 użytkowników korzystających z tego samego serwera pocztowego (tej samej domeny). Jak już wcześniej było wspomniane dzieje się to przy nawiązaniu 1 połączenia.

Argument <forward-path> może zawierać więcej niż jeden adres skrzynki odbiorczej. Ze względów historycznych mogą się tutaj pojawić również lista hostów źródłowych i skrzynka odbiorcza, jednak klienty SMTP (jak już wcześniej było wspomniane) nie powinny korzystać z tej możliwości. Serwer jednak musi być przygotowany na taką ewentualność, ale powinien zignorować tą listę.

Dodatkowe parametry (<rcpt-parameters>) będą omówione później.

Trzecim krokiem w trakcie transakcji jest polecenie DATA. Składnia jego jest prosta po prostu

DATA <CRLF>

Jeżeli polecenie zostało zaakceptowane serwer SMTP zwraca status 354 i prosi klienta aby odpowiedział jak najszybciej treścią maila, kończąc transmisję znacznikiem końca maila. Przykładowo

354 End data with <CR><LF>.<CR><LF>

Kiedy serwer otrzyma znak końca wiadomości, a mail zostanie zapisany odbiorca maila odpowiada standardowo 250 OK. Koniec wiadomości klient zaznacza poprzez przesłanie linii zawierającej jeden znak "." (kropkę). Występują dodatkowo odpowiednia procedura zabezpieczająca nieporozumienie związane z tym, że przy przekazywaniu tekstu wiadomości klient prześle linie zawierającą jedynie znak kropki "." (4.5.2 sekcja).

Polecenie DATA może nie udać się tylko w 2 przypadkach:

- Jeżeli nie było polecenia MAIL albo RCPT albo oba zostały odrzucone wtedy serwer na DATA możę odpowiedzieć

  503 Command out of sequence

albo też
554 No valid recipients
Jeżeli któraś z tych odpowiedzi (albo inna z serii 5yz czyli mówiących o błędzie) zostanie odebrana przez klienta, nie powinien on przesyłać treści wiadomości. Powinien on robić to tylko w przypadku odpowiedzi 354.

- Jeżeli polecenie DATA zostało zaakceptowane odpowiedzią 354, może dojść do nieudanego przesłania maila tylko gdy transakcja jest niekompletna (brak zdefiniowancyh odbiorców), albo niespodziewanie odbiorca przestanie być dostępny lub też serwer z innych powodów (np. polityka ochrony) serwer postanowi odrzucić wiadomość.

W praktyce jednak niektóre serwery nie dokonują weryfikacji odbiorcy przed otrzymaniem całej wiadomości. Nie jest to dobre zachowanie, gdyż jeżeli po zaakceptowaniu adresów odbiorców nagle po poleceniu DATA serwer odpowie "550 mailbox not found" to klient nie jest w stanie określić, który adresat jest nieprawidłowy.

Serwer nie powinien na tym etapie odrzucać wiadomości na podstawie nagłówków przekazanych w treści wiadomości (po poleceniu DATA).

Oto cała przykładowa, prawidłowa sesja protokołu SMTP:
Klient otwiera połączenie do serwera.
S: 220 smtp.example.com ESMTP Postfix
C: HELO relay.example.org
S: 250 Hello relay.example.org, I am glad to meet you
C: MAIL FROM:<bob@example.org>
S: 250 Ok
C: RCPT TO:<alice@example.com>
S: 250 Ok
C: RCPT TO:<theboss@example.com>
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: From: "Bob Example«bob@example.org>
C: To: Alice Example <alice@example.com>
C: Cc: theboss@example.com
C: Date: Tue, 15 Jan 2008 16:02:43 -0500
C: Subject: Test message
C:
C: Hello Alice.
C: This is a test message with 5 headers and 4 lines in the body.
C: Your friend,
C: Bob
C: .
S: 250 Ok: queued as 12345
C: QUIT
S: 221 Bye
Serwer zamyka połączenie.

3.2.4 Polecenia do debugowania adresów

SMTP dostarcza poleceń za pomocą których można dokonać weryfikacji nazwy użytkownika albo uzyskać zawartość listy mailingowej. Dokonuje się tego przy pomocy poleceń VRFY i EXPN, których wspacie powinno być zaimplementowane w serwerze SMTP.

Argumentem polecenia VRFY jest po prostu napisa, który składa się z nazwy użytkownika lub też dodatkowo domeny. Jeżeli w wyniku polecenia serwer zwróci odpowiedź 250, może zawrzeć w nim pełną nazwę użytkownika i musi zawrzeć skrzynkę użytkownika. Musi więc być to jedno z poniższych:

- User Name <local-part@domain>
- local-part@domain

Kiedy argument podany w komendzie VRFY składa się z kilku adresów, serwer może zgłosić niejasność w adresie jakos całości albo wskazać na jeden z podanych w argumencie adresów. Dla przykładu odpowiedzi serwera na komendę VRFY mogą przyjąć postać:

- 530 User ambiguous
- 553- Ambiguous; Possibilities are 553-Joe Smith <jsmith@foo.com> 553-Harry Smith <hsmith@foo.c
- 553-Ambiguous; Possibilities 553- <jsmith@foo.com> 553- <hsmith@foo.com> 553 <dweep@foo.com

W normalnych okolicznościach klient, który otrzyma odpowiedź 553 oczekuje, że zwrócone wyniki będzie mógł zaprezentować użytkownikowi czyli będą one dla niego zrozumiałe. W powyższym przykładzie możliwe jest to tylko w przypadku 2 i 3 (pierwszy nic konkretnego poza informacją o błędzie nie mówi).

Drugie polecenie EXPN przyjmuje jako argument napis (string), który identyfikuje listę maili, a odpowiedź poprawna serwera może zawierać pełne nazwy użytkowników i musi podawać ich skrzynki mailowe.

Na niektórych hostach występuje problem z rozróżnieniem pomiędzy listą mailową, a aliasem do pojedynczej skrzynki mailowej, ponieważ struktura danych przechowująca informacje na ten temat może przyjmować oba typy wpisów i jest na przykład możliwe, że lista mailowa zawiera tylko jeden adres skrzynki. Próba weryfikacji takiego adresu poprzez VRFY, może udać się jedynie gdy serwer SMTP umie obsłużyć wiadomość wysłaną na ten adres poprzez dostarczenie jej do każdej skrzynki z listy. Jeżeli nie jest on w stanie tego zrobić zwróci jeden z następujących błędów

550 That is a mailing list, not a user lub też 252 Unable to verify members of mailing list

W przypadku wielolinijkowej odpowiedzi serwera na polecenie EXPN dokładnie jedna skrzynka mailowa powinna znaleźć się w lini.

Przykładowe polecenie EXPN i odpowiedzi na nie:

C: EXPN Example-People S: 250-Jon Postel <Postel@isi.edu> S: 250-Fred Fonebone <Fonebone@phys

C: EXPN Executive-Washroom-List S: 550 Access Denied to You.

Serwerowi nie wolno w odpowiedzi na VRFY lub EXPN zwracać odpowiedzi 250 dopóki nie dokona faktycznej weryfikacji adresu. Są jednak przypadki, w których adres wydaje się być prawidłowy, ale w czasie rzeczywistym nie może on

zostać zweryfikowany. Jest tak np. kiedy serwer służy jako pośredni dla innego serwera albo domeny. W takim przypadku serwer dokonuje jakby "pozornej weryfikacji". Sprawdza on poprawność składniową adresu, jak również może dokonać sprawdzenia domeny czy jest ona w zakresie domen do których maile może przekazywać. Odpowiedź serwera powinna więc uwzględnić, że jest to tylko częściowa weryfikacja adresu i zamiast statusu 250 powinien pojawić się status 252.

Implementacje serwera SMTP powinny zawierać przedstawione powyżej komendy. W celach bezpieczeństwa implementacje mogą dostarczać narzędzi (np. poprzez plik konfiguracyjny) do wyłączenia tych poleceń. Polecenia te były opcjonalne w RFC 821, ale obecnie powinny być wylistowane jako dodatkowe serwisy w odpowiedzi na EHLO.

3.2.5 Koniec sesji i połączenia

Jak już wcześniej zostało wspomniane sesja w protokole SMTP kończy się wraz z przesłaniem przez klienta polecenia QUIT. Serwer odpowiada pozytywnym kodem, po którym zamyka połączenie.

Serwerowi nie wolno zamknąć połączenia intuicyjnie poza 2 przypadkami:

- otrzyma polecenie QUIT i odpowie kodem 221
- po wykryciu potrzeby zamknięcia serwisu SMTP i przesłaniu kodu 421. Odpowiedź ta może być umieszczona po każdej komendzie lub asynchronicznie (przy założeniu, że klient odbierze ją zanim kolejne polecenie zostanie przesłane)

Klient SMTP, który spotka się z zamknięciem połączenia w wyniku okoliczności, których nie może obsłużyć, powinien traktować przeprowadzaną transakcję jak gdyby otrzymał od serwera odpowiedź 451 "Requested action aborted: error in processing".

3.2.6 Listy mailowe (mailingowe) i aliasy

Serwer SMTP powinien wspierać zarówno aliasy, jak i listy mailowe, które służą w celach dostarczenia wiadomości do większej ilości adresów.

Alias to inaczej pseudo-skrzynka, której nazwa przy odbieraniu maila tłumaczona jest na rzeczywisty adres (bądź adresy). Zmieniane jest tylko pole (RCPT TO), reszta wiadomości ("koperta" i ciało wiadomości) pozostaje nietknięta.

Lista mailowa zaś służy dostarczoniu albo przekazaniu maila do wszysktich adresów znajdujących się na liście. W wyniku tej operacji adres zwrotny na "kopercie" (MAIL FROM) musi być zmieniony na adres osoby albo innej jednostki, która sprawuje opiekę nad listą. Nagłówki wiadomości (w szczególności pole From) pozostają nietknięte.

3.2.7 Informace o trasie

Kiedy serwer SMTP otrzyma wiadomość do dostarczenia albo dalszego procesowania pozostawia w niej swój ślad. Ślad ten umieszczany jest na początku ciała wiadomości (nagłówki "time stamp" albo "Received").

Linia ta powinna się składać z następujących informacji:

- pola FROM, które powinno zawierać zarówno nazwe hosta źródłowego (tak ja w zaprezentowanym w ramach odpowiedzi na komende EHLO) i adres IP źródła, zdeterminowane z połączenia TCP
- pola ID, które może zawierać zgodnie z sugestiami w RFC 822 znaczke "@", ale nie jest to wymagane
- pola FOR, które może zawierać listę wpisów <path> kiedy w poleceniu RCPT zostało podanych ich kilka

Wcześniej dodane linijka Received nie powinna być zmieniana. Serwery SMTP powinny mieć przygotowaną linie Received do "wstrzyknięcia" do wiadomości. Nie wolno im zmieniać obecnych lini ani też wpisywać swojej lini w innym niż przeznaczone miejscu.

Przykładowa linia wygląda następująco:

```
Received: from mizar.astronet.pl ([127.0.0.1]) by localhost (mizar [127.0.0.1]) (amavisd-new, port 10024
```

W miarę rozwoju Internetu linie Received są bardzo istotne w celach wykrywania wystąpienia jakichkolwiek problemów.

Kiedy wiadomość jest ostatecznie dostarczana, serwer SMTP wstawia nagłówek "Return-path" na początku maila. Zachowanie to jest wymagane. Systemy mailowe SMTP muszą to implementować. "Return-path" to zachowane informacje z argumentu <reverse-path> z polecenia MAIL. Po tym oznaczeniu uznawane jest, że wiadomość opuściła środowisko SMTP, jednakże nie oznacza to obecnie, że została ona ostatecznie dostarczona. Może być transmitowana przez inny system mailowy.

3.2.8 Rozmiary i opóźnienia

W protokole SMTP występują obiekty, które mają wymagane minimalne bądź też maksymalne rozmiary. Każda implementacja musi być w stanie odbierać o takich wielkościach. Obiekty większe powinny być unikane. Klient może liczyć się z tym, że przy próbie transmisji zbyt dużych obiektów, może spotkać się z odmową ze strony serwera.

Obiekty o których mowa to:

- local-part - maksymalna długość nazwy użytkownika. Wynosie ona 64 znaki
- domain - maksymalna długość domeny. Wynosi to 255 znaków
- path - maksymalna długość argumentu reverse-path albo forward-path. Wynosi ona 256 znaków (wliczając w to interpunkcje i separatory elementów)
- comman line - maksymalna długość polecenia wraz z jego nazwą i znakiem <CRLF>. Wynosi ona 512 znaków. Rozszerzenia SMTP mogą posłużyć w celach zwiększenia tego limitu
- reply line - maksymalna długość lini odpowiedzi. Wynosia oan 512 znaków wliczając w to kod odpowiedzi oraz znak <CRLF>

- text line - maksymalna długość lini tekstu wliczając w to znak <CRLF>. Wynosi ona 1000 znaków. Może być ona rozszerzona przez dodatkowe serwisy SMTP.
- message content - długość zawartości wiadomości (wliczając w to nagłówki wiadomości jak i jego ciało). Musi mieć ona co najmniej 64K octets (czyli ile?). Odkąd wprowadzono standard Multipurpose Internet Mail Extensions (MIME) wielkość wiadomości wzrosła diametralnie, dlatego też zaleca się obecnie aby serwery SMTP nie nakłady żadnych ograniczeń na wielkość wiadomości
- recipients buffer - rozmiar buffora adresowego. Minimalna całkowita liczba odbiorców, która musi być zbuforowana to 100. Klient, który chce dostarczyć wiadomość do więcej niż 100 odbiorców powinien być przygotowany na transmisję maila w kawałkach po 100 odbiorców (jeżeli oczywiści serwer akceptuje wiadomości z taką ilością odbiorców)

Błędy związane z powyższymi ograniczeniami mogą być zgłoszone poprzez standardowe kody błędów. Przykłady takich błędów:

- 500 Line too long
- 501 Path too long
- 452 Too many recipients
- 552 Too much mail data

W ramach protokołu SMTP nie możemy także zapomnieć o opóźnieniach, które należy wziąć pod uwagę przy implementacji klienta. Klient musi repektować opóźnienia związane z poleceniami. Powinny one być w prosty sposób w kliencie konfigurowalne (w celach ewentualnych zmian i próby dostosowanie opóźnień do warunków panujących w sieci). Proponowane opóźnienia w kliencie:

- Początkowa wiadomość 220 od serwera: 5 minut
  Klient SMTP powinien mieć możliwość rozróżnienia pomiędzy problemami związanymi z TCP (zerwane połączenie) a opóźnieniami w otrzymaniu powitalnego 220. Wiele serwerów SMTP akceptuje połączenia, ale opoźnia ich obsługę dopóki load na maszynie nie spadnie do odpowiedniego poziomu
- Polecenie MAIL: 5 minut
- Polecenie RCPT: 5 minut
  Większe opóźnienie jest wymagane w przypadkach, gdy procesowanie obsługi listy mailowej czy aliasu nie następuje po akceptacji maila
- Inicjalizacja polecenie DATA: 2 minuty
  Potrzeba na czekanie na odpowiedź serwera "354 Start Input"
- Blok danych: 3 minuty
  Ewentualna potrzeba na oczekiwanie na zakończenie wywołanai TCP SEND do transmisji kolejnych porcji danych

- Zakończenie polecenia DATA: 10 minut
  Potrzebne w celach oczekiwania na odpowiedź 250 OK. Kiedy odbiorca otrzyma ostatni znak oznaczający koniec wiadomości, przeważnie zaczyna on procesowanie dostarczenia wiadomości do skrzynku użytkownika. Nieodpowiednie opóźnienie tutaj może skutkować dostarczeniem paru tego samego maila, ponieważ klient założy, że transakcja się nie udała (minie odpowiedni czas), a tak naprawde odbiorca zaakceptuje wiadomość, a jeszcze nie zdąży powiadomić o tym klienta.

W przypadku opoźnień serwera możemy mówić o stałym opóźnieniu 5 minut w oczekiwaniu na komendy od klienta.

3.2.9 Strategie wysyłania i odbierania

Przeważnie struktura implementacji hosta SMTP zawiera skrzynki użytkowników, jedną lub więcej kolejek wiadomości oraz jednego lub więcej demonów do odbierania i wysyłania maili. Dokładna struktura różni się w zależności od potrzeby użytkowników oraz liczby i wielkości list mailowych obsługiwanych przez host. Poniżej strategie, które okazały się bardzo pomocne w przypadku obsługi dużego ruchu mailowego:

- Wysyłanie:
  Głównym modelem dla klienta SMTP jest jeden lub kilka procesów, które cyklicznie próbują transmitować maile wychodzące. W typowym systemie, program, który komponuje wiadomość posiada odpowiednie metody, aby zwrócić natychmiastową uwagę na nowy element w wiadomościach wychodzących. Maile, które nie są transmitowane od razu są kolejkowane i cyklicznie wysyłane. Kolejka mailowa zawiera oprócz samej wiadomości "kopertę" czyli dodatkowe informacje służące do wysyłania.
  Nadawca po nieudanym wysłaniu powinien odczekać pewną jednostkę czasu zanim ponowi próbę transmisji maila. Generalnie jednostka ta powinna wynosić około 30 minut.
  Ponawianie wysłania następuje dopóki zakończy się ono sukcesem bądź też klient się podda. Przyjmuje się, że po 4-5 dniach nieudanych próbach, wiadomość zostaje uznana za niedoręczoną.
  Klient powinien trzymać listę hostów do których nie może dotrzeć i zapamiętywać opóźnienia do nich, zamiast nieustannie kolejkować maile przeznaczone do tych hostów. W ten sposób szybciej odrzuci maile, które nie mają szansy osiągnąć celu.
  Klient SMTP może zmniejszyć oczekiwanie w kolejce we współpracy z serwerem SMTP. Dla przykładu, jeżeli mail z danego adresu (domeny), został odebrany prawdopodobne jest, że maile kolejkowane dla tego adresu (domeny), mogą zostać wysłane.
  Klient SMTP może mieć kolejkę wielu wiadomości dla każdego nieosiągalnego hosta docelowego. Jeżeli każda z tych wiadomości w cyklu ponawiania będzie wysyłana system zostanie zablokowany na długi okres czasu, ponieważ klient może stwierdzić, że dostarczenie się nie udało jedynie po określonym czasie (do kilku minut). Nawet zmniejszenie opóźnienia do 1 minuty przy setkach wiadomości powoduje zamrożenie systemu.
  W przypadku wysyłania wiadomości do wielu odbiorców z tego samego

hosta klient powinien stosować taktykę MAIL, RCPT, RCPT, ... RCPT, DATA zamiast MAIL, RCPT, DATA, ..., MAIL, RCPT, DATA.

- Strategie odbioru
  SMTP server powinien oczekiwać nieustannie nadchodzących połączeń. Wymaga to oczywiście od implementacji obsługi wielu połączeń jednocześnie. Jak już wspomniano wcześniej obsługa procesu dostarczenia maila do konkretnej skrzynki może odbywać się nie tuż po obsłudze polecenia DATA, ale w odzielnym procesie. Zmniejszony jest wtedy czas oczekiwania klienta na ostateczne potwierdzenie zakończenia transakcji maila.

3.2.10 Rozwiązywanie adresu i przekazywanie maila

Zanim klient SMTP zidentyfikuje dokąd ma być posłany mail, musi nastąpić odpytanie serwera DNS o adres domeny. Oczekuje się, że podane w adresie nazwy domen są w pełni akceptowanymi nazwami domen (fully qualified domain name, FQDN). Pierwsze odpytanie serwera to zapytanie o rekorkdy MX związane z nazwą.

Rekord MX (Mail exchanger) to jeden z typów rekordów w serwerze domen DNS, który specyfikuje w jakis sposób maile powinny być routowane (przekazywane w sieci) przy użyciu protokolu SMTP. Każdy rekord MX ma następującą postać: [name] [ttl] IN MX preference host gdzie:

- name - nazwa komputera lub domeny czyli cel ostateczny naszego maila
- ttl - czas życia rekordu
- preference - komputer lub domena może mieć więcej niż jeden komputer wskazany jako odbiorca poczty, wpisana w tym polu liczba określa preferencję - im mniejsza jej wartość tym priorytet wyższy
- host - nazwa serwera pocztowego

Jeżeli przed rekordem MX zostanie znaleziony rekord CNAME, host który znajduje się pod tym rekordem jest następnie traktowany jak host, który był pierwotnie podany do poszukiwań w DNS-ie. Rekord CNAME to swego rodzaju alias w DNS-ie.

Jeżeli żaden rekord MX nie zostanie odnaleziony, ale za to odnaleziony zostanie rekord A, jest on traktowany jakby był rekordem MX z preferencją 0 (czyli najwyższym priorytetem).

Jeżeli uda się znaleźć jeden lub więcej rekordów MX dla danej nazwy, implementacjom systemów SMTP nie wolno używać żadnych rekordów A związanych z nazwą.

Kiedy odpytanie DNS-ów zostanie zakończone sukcesem, możemy w wyniku mapowania otrzymać kilka adresów, ze względu na możliwość kilku rekordów MX czy też multihoming (albo to i to). Ażeby zapewnić niezawodną transmisję maila, klient SMTP musi próbować (i ew. ponawiać próby) dla każdego z adresów wg porządku, aż uda się dostarczyć wiadomość. Klient może posiadać konfigurowalną liczbę maksymalnych alternatywnych adresów. Liczba ta powinna wynosić przynajmniej 2 adresy.

Jak wyżej wspomniano klient posyła maile do kilku adresów na podstawie uporządkowanej listy, która to zależna jest od priorytetów w rekordach MX oraz

od hostów z wieloma adresami. Im niższy priorytet tym serwer jest bardziej preferowany, zaś przy tych samych priorytetach dokonywany jest wybór losowy.

Adres hosta docelowego może okazać się nie pojedynczym adresem, ale listą adresów (multihomed host). Zadaniem serwera DNS jest zwrócenie już uporządkowanej listy adresów IP, a klient po kolei próbuje wysłać pod nie maile.

Obsługa alternatywnych adresów związanych z multihomingiem może zostać w kliencie ograniczona do określonej liczby albo w ogóle wyłączona.

3.3 Konstrukcja wiadomości

Główne informacje na temat wyglądu wiadomości są obecnie zebrane w kilku dokumentach. Jednym z głównych opisujących wygląd wiadomości bez rozszerzeń jest obecnie RFC 2822 (następca już uznanego za przedawniony RFC 822). Rozszerzenia wiadomości, które będą opisane później, zdefiniowane są w kolejnych dokumentach (seria dokumentów o MIME RFC2045, RFC2046, RFC2047, RFC2048, RFC2049).

Sama wiadomość, z punktu widzenia mocno niskopoziomowego, to po prostu seria znaków. Wiadomość zgodna ze standardem składa się ze znaków od 1 do 127 interpretowanych jako znaki kodowania US-ASCII. Wynika to oczywiście ze względów historycznych, gdyż, jak to juz było wspomniane, to w Stanach zostały posłane pierwsze wiadomości i to na tamte potrzeby stworzono pierwsze standardy.

Wiadomości są podzielone na linie. Linia to seria znaków, która jest ograniczona przez 2 znaki powrót karetki (CR - carriege return - w ASCII znak numer 13) i znak nowej linii (LF - line-feed - w ASCII znak numer 10). Znaki te z reguły występują razem w ramach wiadomości i oznaczane są jako CRLF.

Ogólnie wiadomość można podzielić na 2 części - pola z nagłówkami wiadomości (nazywane po prostu nagłówkiem wiadomości) oraz ciała wiadomości, które jest opcjonalne.

3.3.1 Ograniczenia w długości linii

Ilość znaków w linii jest ograniczona dwoma limitami. Maksymalna ilość znaków w linii (z pominięciem CRLF), musi być mniejsza niż 998 znaków, i nie powinna być większa niż 78 znaków.

Limit 998 znaków wynika z ograniczeń wielu obecnych implementacji, które zajmują się odbiorem, wysyłką oraz przechowywaniem wiadomości. Nie obsługują one po prostu dłuższych linii.

Bardziej konserwatywne ograniczenie związane z liczbą 78 znaków wynika z próby dostosowania wiadomości do wielu implementacji interfejsu użytkownika, które wyświetlają te wiadomości. Implementacje te mogą uciąć linie zawierające więcej niż 78 znaków.

3.3.2 Nagłówki wiadomości

Pole nagłówka składa się z:

- nazwy nagłówka po której następuje dwukropek ":"
- ciała nagłówka po które następuje znak CRLF

Nazwa nagłówka musi składać się z drukowalnych znaków US-ASCII (czyli tych, których kody są pomiędzy 33, a 126 włącznie) bez przecinka ",". Ciało nagłówka może zawierać wszystkie znaki oprócz CR i LF, chyba, że przy konstrukcji nagłówka użyto składania (ang. folding) opisanego niżej.

Same nagłówki można podzielić na posiadające (structered) lub nie posiadające dodatkowej struktury (unstructered).

Te drugie (unstructered) oprócz wcześniej nałożonych restrykcji co do występowania określonych znaków, nie mają dodatkowych obostrzeń. Semantycznie nie poddaje się ich dodatkowemu procesowaniu (chyba, że nagłówek należy poskładać [ang. unfold])

Niektóre nagłówki wymagają w ramach swojego ciała dodatkowej semantyki (struktury). Jeżeli dane ciało nagłówka nie jest zgodne z tą semantyką to taki nagłówek, a przez to także wiadomość, nie jest zgodna ze specyfikacją opisywaną w RFC 2822.

Jak już było wcześniej zaznaczone pojedynczy nagłówek to pojedyncza linia składająca się z nazwy, dwukropka i ciała nagłówka. Dla wygody, a także aby poradzić sobie z ograniczeniem na maksymalna długość linii, pojedynczy nagłówek może być rozłożony na kilka linii. Dzielenie to określamy angielskim słowem "folding". Dzielenie to polega na tym, że ciało nagłówka może zostać, w przypadku długiej lini, przeniesione do następnej. Linia kończona jest normalnym znakiem CRLF, ale nowa linia musi być rozpoczęta znakiem WSP (white space czyli spacją space [SP] - kod 32 lub tabulatorem horizontal-tab [HTAB] - kod 11). Przykładowo nagłówek:

Subject: This is a test

może być reprezentowany jako

Subject: This
 is a test

Dzielenie nagłówka na wiele lini może nastąpić w wielu miejscach jednak zaleca się wstawianie znaku CRLF pomiędzy tokenami, które znajdują się na najwyższym stopniu w semantyce nagłówka. Dla przykładu kiedy ciało nagłówka składa się z podzielonych przecinkami wartości, zaleca się łamanie lini po przecinku rozdzielającym struktury.

Jak już wcześniej było zaznaczone składanie nagłówka z wielu lini określane jest angieslkim słowem "unfolding".

3.3.3 Opis niektórych nagłówków

Niektóre z nagłówków wiadomości (jak to juz wcześniej wspomniano) posiadają pewną dodatkową strukturę i są przeznaczone do określonych celi. Dokładny opis semantyki ciała "zaawansowanych" nagłówków umieszczony jest w RFC 2822 i nie będzie tutaj poruszany, jednak warto wspomnieć jakie informacje niosą te nagłówki.

Date - nagłówek ten specyfikuje datę i czas, którą to twórca wiadomości wskazuje, że jest ona skończona i gotowa do wejścia w system dostarczania

poczty (czyli gotowa do wysłania). Dla przykładu może być to czas kiedy użytkownik tworząc maila klika przycisk "wyślij"

From, Sender, Reply-to - nagłówki określane są nagłówkami autora wiadomości. Pola te wskazują na skrzynkę (skrzynki) źródła wiadomości. W szczególności pole "From"

3.3.4 Ciało wiadomości

Ciało wiadomości to po prostu linie złożone ze znakół US-ASCII. Są tylko dwa wymogi co do znaków w ciele wiadomości:

- znaki CR i LF muszą występować razem w postaci CRLF
- długość lini musi być nie większa niż 998 znaków i powinna być nie większa niż 78 znaków (wyłączając CRLF)

3.3.5 Rozszerzenie wiadomości, MIME

3.4 Obecne wykorzystanie protokołu i jego forma

3.4.1 Kwestie bezpieczeństwa, rozszerzenia protokołu

4 Dzisiejsze narzędzia do filtracji protokołu SMTP

Dzisiejsze

4.1 Konieczność wprowadzenia filtracji

Konieczność

4.2 Produkty komercyjne

Produkty komercyjne

4.2.1 Clearswift

Clearswift

4.2.2 Aladdin eSafe

Aladdin eSafe

4.2.3 Surfcontrol Email Filter

Suontrol Email Filter

4.3 Produkty open-source

5 Opracowany filtr poczty SMTP

5.1 Założenia projektu

5.2 Moduły projektu

5.2.1 Parser wiadomości

5.2.2 Parser reguł

5.2.3 Analizator wiadomości

5.2.4 Kolejka

5.3 Kompilacja, konfiguracja i uruchomienie projektu

5.4 Testy wydajnościowe

6 Spostrzeżenia, wnioski

Literatura

[Cro82] David H. Crocker. *RFC822 - STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES*. 1982.