Network Working Group
Request for Comments: 5321
Obsoletes: 2821
Updates: 1123
Category: Standards Track

J. Klensin
October 2008

Простой протокол передачи электронной почты (SMTP)

Simple Mail Transfer Protocol

Статус документа

Это документ содержит проект стандарта протокола Internet для сообщества Internet и служит приглашением к дискуссии в целях развития и совершенствования протокола. Текущее состояние стандартизации и статус протокола можно узнать из текущей версии документа Internet Official Protocol Standards (STD 1). Допускается свободное распространение документа.

Аннотация

Этот документ является спецификацией базового протокола доставки электронной почты Internet. Документ консолидирует, обновляет и проясняет несколько предшествующих документов, отменяя их полностью или частично. Документ включает механизмы расширения SMTP и обобщение накопленного опыта для современного состояния Internet, но не содержит деталей отдельных расширений протокола. Хотя протокол SMTP разработан для обеспечения почтового транспорта и доставки, в данной спецификации содержится также информация, которая важна для протокола подачи сообщений в пользовательских почтовых агентах чтения почты и мобильных средах.

Оглавление

1. Введение

1.1. Транспортировка электронной почты

Целью протокола SMTP[1] является обеспечение надёжной и эффективной доставки электронной почты.

Протокол SMTP не зависит от конкретных подсистем передачи и требует для работы лишь канал с гарантированной и упорядоченной доставкой потока данных. Хотя в этом документе обсуждается использование транспорта TCP, возможно использование и других транспортных протоколов. Описания некоторых протоколов этого типа даны в приложениях к RFC 821 [1].

Важным свойством протокола SMTP является возможность транспортировки почты через множество сетей, которые обычно называют «почтовыми трансляторами SMTP»[2] (см. параграф 3.6). Сети состоят из обоюдно доступных по протоколу TCP хостов публичной сети Internet, обоюдно доступных по TCP хостов приватных сетей TCP/IP, находящихся на межсетевыми экранами, или хостов некоторых иных локальных и распределенных сред, использующих на транспортном уровне протоколы, отличные от TCP. Используя протокол SMTP, процесс может передавать почту другому процессу в той же сети или некоторых других сетях через трансляторы или шлюзы, доступные из обеих сетей.

Таким путём почтовые сообщения можно передавать через множество промежуточных трансляторов (relay) или шлюзов на пути между отправителем и конечным адресатом. Для определения следующего промежуточного получателя (next-hop) на пути к адресату используется механизм Mail eXchanger (MX) системы доменных имён (RFC 1035 [2], RFC 974 [12], раздел 5 данного документа).

1.2. Предыстория и контекст документа

Документ содержит спецификацию базового протокола передачи электронной почты в сети Internet. Документ консолидирует, обновляет и разъясняет перечисленные ниже спецификации, не изменяя их функциональности:

- исходная спецификация SMTP (Simple Mail Transfer Protocol) - RFC 821 [1];
- требования к системе доменных имён и её использованию для передачи электронной почты - RFC 1035 [2] и RFC 974 [12];
- пояснения и вопросы применимости RFC 1123 [3];
- материалы из механизмов расширения SMTP Extension - RFC 1869 [13];
- редакторские правки и разъяснения к RFC 2821 [14] для повышения статуса до Draft Standard.

Данный документ отменяет действие RFC 821, RFC 974, RFC 1869, RFC 2821 и обновляет RFC 1123 (замена материалов по доставке электронной почты в RFC 1123). Однако в RFC 821 приведены спецификации некоторых свойств, которые не стали достаточно значимыми в среде Internet середины 1990-х, и (в приложениях) некоторые дополнительные транспортные модели. Эти разделы опущены здесь с целью снижения объёма документа, интересующиеся читатели могут обратиться к RFC 821.

Документ также включает некоторые дополнительные материалы из RFC 1123, которые требовалось усилить. Такие материалы определены разными путями - прежде всего просмотром популярных списков рассылки и телеконференций, а также идентификацией проблем и разночтений, которые появлялись в разрабатываемых расширениях SMTP. В тех случаях, где настоящая спецификация выходит за пределы консолидации сведений из более ранних документов и реально отличается от них, приведённые здесь сведения имеют более высокий приоритет, как в техническом, так и в текстовом отношении.

Хотя SMTP разрабатывался как протокол транспортировки и доставки электронной почты, данная спецификация содержит также информацию, которая важна для использования протокола в качестве средства представления сообщений[3], как рекомендовано спецификациями протоколов POP[4] (RFC 937 [15], RFC 1939 [16]) и IMAP (RFC 3501

[1]Simple Mail Transfer Protocol - простой протокол передачи электронной почты.
[2]SMTP mail relayin.
[3]mail submission.
[4]Post Office Protocol - протокол «почтового отделения».

[17]). В общем случае предпочтительно использовать для представления почтовых сообщений отдельный протокол, описанный в RFC 4409 [18]; далее в документе этот вопрос рассматривается более подробно.

В параграфе 2.3 приводятся определения специфических для данного документа терминов. За исключением тех случаев, когда исторически сложившаяся терминология нужна для понимания, в данном документе применяются современные трактовки терминов «клиент» и «сервер» для обозначения передающих и принимающих процессов SMTP, соответственно.

В дополнительном документе RFC 5322 [4] обсуждаются разделы заголовков и тел почтовых сообщений, а также их структура и форматы.

1.3. Соглашение о терминах

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не нужно** (SHALL NOT), **следует** (SHOULD), **не следует** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с RFC 2119 [5]. Такие термины в документе используются очень аккуратно в целях повышения уровня взаимодействия систем электронной почты. Каждое использование таких терминов должно трактоваться как уровень требований в части соответствия данной спецификации.

Поскольку этот документ имеет длинную предысторию, во избежание возникновения ошибок и заблуждений у читателей, большая часть примеров и упомянутых в них доменных имён заимствована из RFC 2821. Читателям следует быть осторожными с приведёнными примерами и не использовать их напрямую в программном коде или конфигурационных файлах.

2. Модель SMTP

2.1. Базовая структура

Структура SMTP показана на рисунке.

Когда у клиента SMTP имеется сообщение для передачи, он организует двухсторонний канал связи с сервером SMTP. Клиент SMTP отвечает за доставку почтовых сообщений одному или множеству серверов SMTP или возврат сообщения об отказе.

Способ представления почтовых сообщений клиенту SMTP и способ определения клиентом идентификаторов (имён) доменов, в которые данное сообщение будет передаваться, определяются локальными условиями и не рассматриваются в этом документе. В некоторых случаях указанный или определённый клиентом SMTP домен идентифицирует конечного получателя сообщения. В других случаях, когда клиенты SMTP связаны с реализациями протоколов POP (RFC 937 [15], RFC 1939 [16]) или IMAP (RFC 3501 [17]) или клиент располагается внутри изолированной транспортной среды, идентифицированный домен будет определять промежуточного получателя, через которого будут транслироваться все почтовые сообщения. Клиенты SMTP, которые передают весь трафик, независимо от домена адресата, связанного с конкретным сообщением, или не поддерживают очередей для повтора попыток передачи сообщений при неудаче, могут соответствовать данной спецификации, но не обеспечивать всех возможностей. Предполагается, что полнофункциональные реализации SMTP, включая трансляторы, которые используются неполными системами и их получателями, будут поддерживать очереди, повторы и альтернативную адресацию, рассматриваемые в данном документе. Во многих случаях упомянутым выше неполнофункциональным клиентам **следует** использовать протокол представления сообщений (RFC 4409 [18]) взамен SMTP.

Способ, с помощью которого клиент SMTP после определения домена адресата, находит сервер SMTP для передачи сообщения, а также процесс передачи определяются данной спецификацией. Для передачи почты SMTP-серверу клиент SMTP организует двухсторонний канал связи с сервером. SMTP-клиент определяет адрес подходящего хоста, на котором работает сервер SMTP, преобразуя доменное имя получателя в MX-запись (Mail exchanger) промежуточного или конечного хоста-получателя.

Сервер SMTP может быть конечным или промежуточным транслятором[1] (после приёма письма сервер выполняет функции клиента SMTP, пересылая сообщение дальше) или шлюзом[2] (может передавать сообщение дальше, используя отличный от SMTP протокол). Команды SMTP генерируются клиентом SMTP и передаются серверу SMTP. Сервер SMTP передаёт отклики в ответ на команды клиента SMTP.

Иными словами, передача сообщения может осуществляться в один приём путём соединения исходного отправителя SMTP с конечным получателем SMTP или через цепочку промежуточных систем. В любом случае после того, как сервер сообщил об успешном завершении операции в конце передачи почтовых данных, происходит формальная передача ответственности - в соответствии с требованиями протокола сервер **должен** принять на себя ответственность за доставку сообщения или должным образом предоставить отчёт о невозможности доставки (см. параграфы 6.1, 6.2 и 7.8 ниже).

После организации коммуникационного канала и согласования параметров клиент SMTP обычно инициирует почтовую транзакцию. Такая транзакция состоит из последовательности команд, задающих отправителя и получателя сообщения, а также передачи содержимого письма (включая все заголовки и прочие структуры). Если одно сообщение передаётся множеству адресатов, разумно передавать одну копию сообщения для всех получателей, доставка которым осуществляется на один хост или через один промежуточный транслятор.

[1]Relay.
[2]Gateway.

Сервер обеспечивает отклик на каждую полученную команду – отклик может показывать восприятие команды (в таких случаях ожидаются дополнительные команды), а также содержать сообщение о временной или постоянной ошибке. Команды, задающие отправителя или получателей, могут включать поддерживаемые сервером SMTP расширения, описанные в параграфе 2.2. Диалог между клиентом и сервером осуществляется поэтапно (команда – отклик – команда …), хотя можно использовать по взаимному согласию конвейерную обработку (RFC 2920 [19]).

После завершения передачи сообщения клиент может запросить разрыв соединения или инициировать следующую почтовую транзакцию. Кроме того, клиент SMTP может использовать соединение с сервером для доступа к дополнительному сервису типа проверки корректности почтовых адресов или получения адресов из списка рассылок.

Как сказано выше, протокол обеспечивает механизм передачи электронной почты. Эта передача обычно осуществляется непосредственно с хоста отправителя на хост получателя, когда оба хоста используют один транспортный сервис. Если же хосты не подключены к общей транспортной системе, передача осуществляется с использованием одного или нескольких промежуточных серверов SMTP. Сегодня в Internet обычной практикой является представление исходного сообщения промежуточному серверу «представления сообщений»[1], который похож на транслятор, но выполняет некоторые дополнительные функции; такие серверы рассматриваются в параграфе 2.3.10 и RFC 4409 [18]. Промежуточный хост в таких случаях действует как транслятор (SMTP relay) или шлюз в другие среды передачи и выбирается обычно с использованием MX-записей DNS (служба доменных имён).

Обычно промежуточные хосты определяются по записям DNS MX, а не путём явного задания маршрута отправителем (см. раздел 5, приложение C и параграф F.2).

2.2. Модель расширений

2.2.1. Базовые вопросы

В рамках программы, начатой в 1990, приблизительно через 10 лет после выпуска RFC 821, протокол был обновлён за счёт добавления модели «расширения услуг»[2], позволяющей клиентам и серверам согласовать использование общих функций, выходящих за пределы исходной спецификации SMTP. Механизм расширения SMTP определяет способ согласования расширенных возможностей клиента и сервера SMTP; сервер может информировать клиента о поддерживаемых расширениях.

Современные реализации SMTP **должны** поддерживать базовые механизмы расширения. Например, сервер **должен** поддерживать команды EHLO, даже если в нем не реализовано соответствующее расширение, а клиентам **следует** использовать команду EHLO вместо HELO[3]. В тех случаях, когда для взаимодействия не требуется явное использование HELO, настоящая спецификация всегда рассматривает только команда EHLO.

Протокол SMTP широко распространён и высококачественные реализации обеспечивают высокий уровень устойчивости к ошибкам. Однако сообщество Internet сейчас считает достаточно важными некоторые службы, которых просто не было в момент создания протокола. При добавлении поддержки таких служб должна обеспечиваться возможность приемлемой работы старых реализаций протокола. К числу таких расширений относятся:

- команда EHLO взамен прежней команды HELO;
- реестр расширений сервиса SMTP;
- дополнительные параметры команд MAIL и RCPT;
- возможность замены команд, определённых в данном протоколе (таких, как DATA) при передаче символов, отличных от ASCII (RFC 3030 [20]).

Сильные стороны протокола SMTP обусловлены, прежде всего, его простотой. Опыт использования множества протоколов показывает, что протоколы с меньшим числом опций получают более широкое распространение, нежели усложнённые протоколы.

Каждое расширение, независимо от обеспечиваемых им преимуществ, должно быть тщательно проверено в части его реализации, развёртывания и совместимости. Во многих случаях стоимость расширения сервиса SMTP может многократно превысить достигаемые преимущества.

2.2.2. Определение и регистрация расширений

Реестр расширенных служб SMTP поддерживается агентством IANA. С каждым расширением связано соответствующее ключевое значение EHLO. Каждая дополнительная служба, регистрируемая IANA, должна быть определена на основе стандартного протокола или одобренного IESG экспериментального протокола. Определение должно включать:

- текстовое имя расширенного сервиса SMTP;
- ключевое значение EHLO связанное с этим расширением;
- синтаксис и возможные значения параметров, связанных с ключевым значением EHLO;
- все дополнительные команды SMTP, связанные с расширением (такие команды обычно используются, но не являются обязательными, как и ключевое значение EHLO);
- все новые параметры расширения, связанные с командами MAIL или RCPT;
- описание воздействия поддержки расширения на поведение клиентов и серверов SMTP;
- размер увеличения максимальной длины команд MAIL и/или RCPT сверх заданного настоящим стандартом.

[1] Message submission server.
[2] Service extensions model.
[3] Однако для совместимости со старыми реализациями клиенты и серверы SMTP по-прежнему **должны** поддерживать команды HELO.

Кроме того, все ключевые значения EHLO, начинающиеся с X или x, указывающие на локальные расширения сервиса SMTP, используются только на основе двухсторонних соглашений. Ключевые слова, начинающиеся с X (независимо от регистра) **недопустимо** использовать в регистрируемых расширениях сервиса. И наоборот, ключевые значения, представляемые в отклике EHLO, который не начинается с X, **должны** соответствовать стандарту, проекту стандарта или одобренному IESG экспериментальному расширению SMTP, зарегистрированному IANA. Для соответствующих требованиям стандарта серверов **недопустимо** предлагать начинающиеся с отличных от X символов расширения сервиса, если они не зарегистрированы.

Имена дополнительных команд и параметров подчиняются тем же правилам, что используются для ключевых значений EHLO; в частности, команды, начинающиеся с X, являются локальным расширением и могут использоваться без регистрации и стандартизации. И наоборот, все команды, которые начинаются с символов, отличных от X, должны регистрироваться.

2.2.3. Дополнительные вопросы, связанные с расширениями

Допускаются расширения, которые могут существенно изменять базовые операции SMTP. Текст в остальных параграфах данного документа следует трактовать с учётом этого обстоятельства. В частности, расширения могут изменять минимальные пределы, указанные в параграфе 4.5.3, отменять требование по использованию набора символов ASCII, упомянутое выше, или вводить некие дополнительные режимы обслуживания сообщений.

В частности, если расширение предполагает, что на пути доставки обычно поддерживаются особые возможности данного расширения, а промежуточная система SMTP определяет, что на следующем этапе такие возможности не поддерживаются, эта система **может** выбрать с учётом конкретного расширения и обстоятельств попытку более поздней доставки и/или выбора другого хоста MX. При использовании такое стратегии тайм-аут возврата к формату без расширения (если таковой имеется) **следует** задавать меньше обычного тайм-аута, используемого для возврата почты в случае невозможности доставки (например, если обычный тайм-аут составляет три дня, тайм-аут для попытки передачи почты без использования расширения может составить один день).

2.3. Терминология SMTP

2.3.1. Почтовые объекты

Протокол SMTP обеспечивает транспортировку объектов электронной почты. Каждый объект состоит из конверта (envelope) и содержимого.

Конверт SMTP передаётся как серия протокольных элементов SMTP (см. главу 3). Конверт содержит адрес отправителя (по которому должны возвращаться отчёты об ошибках) и один или более адресов получателей, а также дополнительную информацию для расширений протокола. В силу исторических причин возможно использование вариаций задания адреса возврата (адреса отправителя) в команде MAIL для указания альтернативных режимов доставки; использование таких вариаций в настоящее время осуждается (см. Приложение F и параграф F.6).

Содержимое SMTP передаётся в виде протокольного элемента SMTP DATA и состоит из двух частей – заголовков и тела. Если содержимое соответствует другим современным стандартам, заголовок состоит из набора полей, каждое из которых включает имя заголовка, двоеточие (;) и данные, структурированные в соответствии со спецификацией формата сообщения (RFC 5322 [4]); тело сообщения, при наличии в нем структуры, соответствует спецификации MIME (RFC 2045 [21]). Содержимое является текстовым по своей природе и выражается с использованием набора символов US-ASCII [6]. Хотя расширения SMTP (типа 8BITMIME, RFC 1652 [22]) могут обходить это ограничение для содержимого, заголовки всегда должны кодироваться с использованием набора символов US-ASCII. Два расширения MIME (RFC 2047 [23] и RFC 2231 [24]) определяют алгоритм представления в заголовках символов, не входящих в US-ASCII, с использованием комбинаций символов набора US-ASCII.

2.3.2. Отправители и получатели

В RFC 821 два хоста, принимающие участие в транзакции SMTP, были описаны как SMTP-sender (отправитель) и SMTP-receiver (получатель). В настоящей спецификации используются иные термины, отражающие сложившуюся практику - SMTP client (иногда просто client) и SMTP server (или просто server) для отправителя и получателя, соответственно. Поскольку в режиме трансляции один хост может выступать в качестве клиента и сервера, продолжается использование терминов «получатель» (receiver) и «отправитель» (sender) там, где это нужно для понимания.

2.3.3. Почтовые агенты и хранилища сообщений

В данной спецификации используется современная терминология, устоявшаяся с момента публикации RFC 821. В частности, клиенты и серверы SMTP обеспечивают почтовый транспортный сервис и, следовательно, называются «агентами доставки почты» - АДП (Mail Transfer Agent или MTA). Пользовательские почтовые агенты – ППА (Mail User Agent, MUA или UA) выступают в качестве исходных отправителей и конечных получателей почтовых сообщений. На стороне отправителя ППА может собирать почту от пользователя для передачи её АДП; агент АДП на стороне получателя передаёт почту ППА (по крайней мере, передаёт этому агенту ответственность за доставку почты; например, помещая её в «почтовое хранилище» - message store). Однако, хотя эти термины достаточно точно выражают суть и применимы к другим средам, границы между ППА (MUA) и АДП (MTA) определены недостаточно чётко. Следовательно, читатель должен внимательно относиться к терминологии.

2.3.4. Хост

В рамках данной спецификации термин «хост» обозначает компьютерную систему, подключённую к Internet (или, в некоторых случаях, к частной сети TCP/IP) и поддерживающую протокол SMTP. Хосты обозначаются именами (см. следующий параграф); **не следует** использовать для идентификации хостов полные адреса (см. параграф 4.1.2).

2.3.5. Доменные имена

Доменное имя (или просто домен) состоит из одной или нескольких разделённых точками компонент. В случае использования в качестве адреса домена верхнего уровня, строка доменного имени не содержит точек. Это ведёт к

возникновению требования, более подробно рассмотренного ниже, об использовании при транзакциях SMTP в публичной сети Internet только полных доменных имён (FQDN[1]), что особенно важно при использовании доменов верхнего уровня. Компоненты доменных имён (метки в терминах DNS RFC 1035 [2]) при транзакциях SMTP могут содержать только последовательности букв[2], цифр, дефиса (-) и знака подчёркивания (_) из набора символов ASCII [6]. Доменные имена используются для обозначения хостов и других объектов иерархии доменных имён. Например, доменное имя может указывать на псевдоним (метка CNAME RR) или метку записи MX (Mail exchanger), которая будет использоваться для доставки почты вместо представленного имени хоста. Дополнительные сведения о доменных именах можно найти в RFC 1035 [2] и разделе 5 данной спецификации.

Доменное имя, как описано в данном документе и RFC 1035 [2], представляет собой полное имя (fully-qualified domain name или FQDN). Доменные имена, не являющиеся FQDN, есть ни что иное, как локальные псевдонимы. В транзакциях SMTP появление локальных псевдонимов **недопустимо**.

При использовании доменных имён в SMTP допускаются только полные (FQDN), преобразуемые DNS имена. Иными словами, разрешено использовать имена, которые могут быть преобразованы в записи MX RR или адреса (RR типа A или AAAA, как сказано в разделе 5), а также CNAME RR, псевдонимы которых могут быть преобразованы в MX или адреса. Локальные псевдонимы и неполные имена использовать **недопустимо**. Указанное правило имеет два исключения:

- доменное имя, указываемое в команде EHLO **должно** быть основным именем хоста (именем, преобразуемым в адрес) или, если у хоста нет имени, полным адресом, описанным в параграфе 4.1.3 и дополнительно рассмотренным при обсуждении команды EHLO в параграфе 4.1.4;
- зарезервированное имя почтового ящика postmaster может использоваться в команде RCPT без полного доменного имени (см. параграф 4.1.1.3) и, в случае такого использования, **должно** приниматься.

2.3.6. Буфер и таблица состояний

Сессии SMTP имеют разные состояния и обе стороны аккуратно поддерживают общее представление о текущем состоянии сессии. В этом документе мы будем представлять это состояние виртуальным буфером и таблицей состояний на сервере, которые могут использоваться клиентом (например, клиент может очистить буфер, сбросить таблицу состояния - в результате чего информация из буфера удаляется, а таблица переходит в некое начальное состояние).

2.3.7. Команды и отклики

Команды SMTP и (если расширение сервиса не задаёт иного) данные сообщений передаются от отправителя к получателю через коммуникационный канал в форме строк.

Отклик SMTP представляет собой подтверждение (или отказ), передаваемое в форме строк от получателя к отправителю через коммуникационный канал в ответ на полученную команду. Общей формой отклика является цифровой код результате (отказ или успешное завершение), за которым обычно следует текстовая строка. Коды служат для использования программами, а текст обычно предназначен для человека. В RFC 3463 [25] содержится спецификация дополнительного структурирования текстовых строк, включая использование дополнений и более специфических кодов завершения (см. также RFC 5248 [26]).

2.3.8. Строки

Строка состоит из некоторого (возможно, нулевого) числа символов данных и завершается символами ASCII для возврата каретки (CR - 0Dh) и перевода строки (LF - 0Ah). Последовательность завершения строки в этом документе будет обозначаться <CRLF>. Для реализаций, соответствующих требованиям данной спецификации, **недопустимо** принимать или генерировать в качестве завершения строки любые другие символы или последовательности символов. Серверы **могут** вносить ограничения на длину строк (см. раздел 4).

В дополнение отметим, что использование в тексте отдельных символов CR или LF (не в комбинации <CRLF>) имеет долгую историю проблем в реализациях почтовых систем и приложениях, работающих с электронной почтой. Для клиентов SMTP **недопустима** передача этих символов за исключением тех случаев, когда комбинация символов служит для завершения строки, а в этом случае **должна** применяться только стандартная последовательность <CRLF>.

2.3.9. Содержимое сообщения и почтовые данные

Термины «содержимое сообщения» (message content) и «почтовые данные (mail data) в этом документе являются взаимозаменяемыми и служат для обозначения информации, передаваемой после восприятия команды DATA до завершения передачи. Содержимое сообщения включает заголовки и (возможно структурированное) тело сообщения. Спецификация MIME (RFC 2045 [21]) обеспечивает стандартные механизмы структурирования тела сообщений.

2.3.10. Отправитель, система доставки, транслятор, шлюз

В данной спецификации различаются четыре типа систем SMTP на основе выполняемых ими функций передачи электронной почты. Система-отправитель (SMTP originator) вносит сообщение в Internet или, в более общем случае, в среду транспортного сервиса. Система доставки (delivery) SMTP принимает почту от транспортного сервиса и передаёт её пользовательскому почтовому агенту или размещает в хранилище сообщений, из которого пользовательский агент может взять почту впоследствии. Транслятор (relay) SMTP получает почту от клиента SMTP и передаёт её другому серверу SMTP (для доставки или следующей трансляции) без изменения данных, добавляя лишь трассировочную информацию в заголовок.

Шлюзами (gateway) SMTP называют системы, получающие почту от клиентов из одной транспортной среды и передающие её серверу другой среды. Различия в протоколах и семантике сообщения по разные стороны шлюза могут потребовать преобразования, которое не может быть выполнено трансляторами SMTP. В контексте данной

[1]fully-qualified domain name.
[2] Английского алфавита. *Прим. перев.*

спецификации межсетевые экраны (firewall), переписывающие адреса, следует рассматривать как шлюзы, даже если по обе стороны экрана используется среда SMTP (см. RFC 2979 [27]).

2.3.11. Почтовый ящик и адрес

В данной спецификации термин «адрес» означает текстовую строку, идентифицирующую пользователя, которому предназначено сообщение, или место, в котором почта будет сохранена. Термин «почтовый ящик» (mailbox) обозначает место хранения почты. Обычно эти термины взаимозаменяемы, если не имеет значения разница между местом хранения почты (почтовый ящик) и её конкретным получателем (адрес). Адрес обычно состоит из пользовательской и доменной части. Стандартные соглашения об именах почтовых ящиков предполагают использование формата local-part@domain - современная терминология поддерживает значительно более широкий спектр применений, нежели просто имена пользователей. По этой причине, а также в результате исторической проблемы, связанной с попытками промежуточных хостов менять локальную часть адреса в целях оптимизации, эта часть адреса **должна** интерпретироваться только хостом, указанным в доменной части адреса.

2.4. Общие синтаксические принципы и модель транзакции

Команды и отклики SMTP подчиняются жёстким синтаксическим правилам. Все команды начинаются с «командного глагола» (command verb), а все отклики – с 3-значного цифрового кода. В некоторых командах и откликах за командой или кодом должны следовать аргументы. Некоторые команды не принимают аргументов (после команды), а за некоторыми кодами откликов может следовать произвольный текст. Во всех случаях присутствия текста он отделяется от команды или кода символом пробела. Полные описания команд и откликов приведены в разделе 4.

Регистр символов в командах и значениях аргументов не имеет значения (т. е., TO: и to: в команде RCPT не различаются), однако это правило имеет исключения для локальной части названия почтового ящика (расширения SMTP могут явно указывать чувствительные к регистру символов элементы). Команды, значения аргументов (кроме локальной части имени почтового ящика) и свободный текст **могут** содержать произвольную комбинацию символов верхнего и нижнего регистра. Для локальной части имён почтовых ящиков регистр символов **должен** приниматься во внимание. Следовательно, реализации SMTP **должны** пытаться сохранить регистр символов в локальной части имени почтового ящика. В частности, для некоторых хостов пользователь smith может отличаться от пользователя Smith. Однако использование чувствительных к регистру локальных частей в именах почтовых ящиков снижает уровень взаимодействия – следует избегать такого применения локальных имён. Домены в почтовых адресах соответствуют обычным правилам DNS и, следовательно, не чувствительны к регистру символов.

Некоторые серверы SMTP в нарушение данной спецификации (и RFC 821) требуют от клиентов представления команд в верхнем регистре. В реализациях **могут** приниматься меры для представления команд в соответствии с требованиями таких серверов.

Поле аргументов содержит текстовую строку переменной длины, заканчивающуюся символами <CRLF>. Принимающая сторона не будет предпринимать никаких действий до получения стандартного завершения строки.

Синтаксис каждой команды рассматривается ниже вместе с описаниями команд. Общие элементы и параметры рассмотрены в параграфе 4.1.2.

Команды и отклики состоят из символов ASCII [6]. Когда транспортный сервис обеспечивает 8-битовый (байты или октеты) канал передачи, каждый 7-битовый символ передаётся с выравниванием по правому краю (старший бит октета имеет нулевое значение). Стандартный сервис SMTP обеспечивает поддержку только 7-битовых символов. Клиенту-отправителю SMTP, который не смог согласовать подходящее расширение с сервером (см. следующий параграф), **недопустимо** передавать сообщения, содержащие информацию в старших битах октетов. Если в нарушение этого правила такое сообщение передаётся, принимающий сервер SMTP **может** сбросить старший бит или отвергнуть сообщение как некорректное. В общем случае транслятору SMTP **следует** предполагать, что содержимое принятого сообщения корректно и, в предположении что конверт позволяет это сделать, транслировать сообщение без проверки его содержимого. Конечно, если содержимое некорректно и путь передачи не может его воспринять, такое решение может привести к доставке конечному адресату искажённого сообщения. Системы доставки SMTP **могут** отвергать такие сообщения или возвращать их, как недоставляемые, вместо попытки доставки. В отсутствии предлагаемого сервером расширения, явно позволяющего делать это, никаким передающим системам SMTP не дозволяется передавать envelope-команды, содержащие символы, не включённые в набор US-ASCII; принимающим системам **следует** отвергать такие команды, используя стандартный отклик 500 syntax error - invalid character.

Клиент **может** запросить у сервера передачу 8-битового содержимого сообщений с использованием расширенных возможностей SMTP, прежде всего 8BITMIME RFC 1652 [22]. Серверам SMTP **следует** поддерживать режим 8BITMIME. Однако это **недопустимо** трактовать, как разрешение на неограниченную передачу 8-битовых символов и не позволяет передавать в конвертах сообщений символы, отличные от ASCII. Для отправителя **недопустимо** запрашивать режим 8BITMIME при передачи данных, где в качестве старшего бита не используется соответствующий формат MIME с подходящим транспортным кодирование; серверы **могут** отвергать такие сообщения.

Используемая в этом документе металингвистическая нотация соответствует нотации Augmented BNF, принятой в документах других почтовых систем Internet. Читателям, которые незнакомы с этим синтаксисом, следует прочесть спецификацию ABNF в RFC 5234 [7]. Для ясности термины метаязыка, используемые в тексте, обозначены угловыми скобками (например, <CRLF>). Читателям следует отдавать отчёт в том, выражения метаязыка могут быть неполными. Имеется множество случаев, когда представленная в тексте информация ограничивает или иначе меняет синтаксис или семантику метаязыка.

3. Обзор процедур SMTP

В этом разделе приведены описания процедур, используемых в SMTP: инициирование сеансов, почтовые транзакции, пересылка почты, проверка имён почтовых ящиков, обработка списков рассылки, а также организация и завершение обмена данными. Вопросы трансляции почты, почтовых доменов и смены ролей рассматриваются в конце раздела. В Приложении D рассматривается несколько конкретных сценариев почтовых транзакций.

3.1. Инициирование сеанса

Сеанс SMTP инициируется, когда клиент соединяется с сервером и сервер отвечает соответствующим сообщением.

Реализация сервера SMTP **может** включать идентификацию своих программ и сведения об их версии в отклик подтверждения соединения после кода 220, на практике эта информация позволяет упросить поиск и решение проблем. Реализации серверов **могут** включать возможность запрета передачи данных о программе и её версии в целях безопасности. Хотя некоторые системы указывают свои контактные адреса для связанных с почтой проблем, это не может служить заменой поддержки требуемого стандартом адреса postmaster (см. раздел 4).

Протокол SMTP позволяет серверу формально отвергать транзакцию, не запрещая изначальные соединения: код 554 **может** возвращаться в открывающем сообщении взамен кода 220. Сервер, использующий такой вариант, **должен** по-прежнему ждать, пока клиент передаст команду QUIT (см. параграф 4.1.1.10) перед закрытием соединения, а на любую мешающую команду **следует** возвращать отклик 503 bad sequence of commands (некорректная последовательность команд). Поскольку попытка организации SMTP-соединения с такими системами может приводить к ошибке, серверу, возвращающему код 554, **следует** передавать вместе с кодом информацию, которая позволит передающей системе понять причину ошибки.

3.2. Инициирование клиента

После того, как сервер передал приглашающее сообщение (приветствие) и клиент получил его, последний обычно передаёт серверу команду EHLO, идентифицирующую клиента. В дополнение к открытию сеанса использование EHLO показывает, что клиент способен работать с расширенным сервисом и запрашивает у сервера список поддерживаемых им расширений. Старые системы SMTP, не способные поддерживать расширения сервиса, и современные клиенты, которым не требуется расширенный сервис в инициируемом почтовом сеансе, **могут** использовать HELO взамен EHLO. Для серверов **недопустимо** возвращать расширенные отклики в стиле EHLO в ответ на команду HELO. Для конкретной попытки соединения, если сервер возвращает отклик command not recognized (команда не распознана) на команду EHLO, клиенту **следует** начать процесс заново и передать команду HELO.

Хост, передающий команду EHLO, идентифицирует в ней себя; команду можно интерпретировать как Hello, I am <domain> (Привет, я домен …), а для случая EHLO – and I support service extension requests (и я поддерживаю расширения …).

3.3. Почтовые транзакции

Почтовая транзакция SMTP состоит из трёх этапов. Началом транзакции служит команда MAIL, дающая идентификацию отправителя (в общем случае команда MAIL может быть введена только при отсутствии незавершённых почтовых транзакций; см. параграф 4.1.4.). После этого следует одна или несколько команд RCPT, указывающих получателей сообщения. Последний этап транзакции начинается командой DATA, которая инициирует передачу почтовых данных и завершается индикатором end of mail, который также подтверждает транзакцию.

Первым этапом транзакции является команда MAIL.

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

Эта команда говорит получателю SMTP о начале новой почтовой транзакции и сбрасывает все таблицы состояний и буферы, включая любые данные получателя или почтовые данные. Часть <reverse-path> (обратный путь) первого или единственного аргумента команды содержит название почтового ящика отправителя (между скобками < и >), которое может использоваться для передачи отчётов об ошибках (см. параграф 4.2). Восприняв команду, сервер SMTP возвращает отклик 250 OK. Если указанный почтовый ящик по каким-то причинам неприемлем, сервер **должен** возвратить отклик, показывающий временной тип отказа – постоянная (т. е., повторится при повторе команды клиентом) или временная (т. е., адрес клиента может быть принят при следующем вызове) ошибка. Несмотря на очевидность этого требования, существуют обстоятельства, при которых возможность восприятия обратного пути невозможно определить, пока не будет получен по крайней мере один прямой путь (в команде RCPT). В таких случаях сервер **может** воспринять обратный путь (отклик 250) и сообщить о возникновении проблем после получения и проверки прямых путей. Обычно это делается с помощью откликов 550 или 553.

Исторически <reverse-path> может содержать больше данных, нежели просто имя почтового ящика, но современным системам **не следует** использовать маршрутизацию почты отправителем - source routing (см. Приложение C).

Дополнительные параметры <mail-parameters> связываются с согласованным расширением сервиса SMTP (см. 2.2).

Вторым этапом транзакции является команда RCPT. Данный этап может повторяться много раз.

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

Первый или единственный аргумент этой команды включает прямой путь forward-path (обычно имя почтового ящика и домена, обязательно заключённые в скобки <>), идентифицирующий получателя. Восприняв команду, сервер SMTP возвращает отклик 250 OK и сохраняет прямой путь. Если известно, что почта не может быть доставлена адресату, сервер SMTP возвращает отклик 550, обычно сопровождаемый строкой типа "no such user - " с именем почтового ящика, для которого невозможна доставка (возможны также другие обстоятельства и коды возврата).

Параметр <forward-path> может содержать не только адрес получателя. Исторически <forward-path> может включать маршрут (source routing) к получателю в виде списка промежуточных хостов, однако современным клиентам SMTP **не рекомендуется** использовать маршрутизацию почты отправителем (см. Приложение C). Сервер **должен** быть готов к восприятию списка source route в прямом пути, но **рекомендуется** игнорировать эти маршруты и **можно** отклонять предлагаемую таким маршрутом трансляцию. Подобно этому, сервер **может** отказаться от приёма почты, предназначенной для других хостов или систем. Эти ограничения делают сервер бесполезным в качестве транслятора для клиентов, не полностью поддерживающих функциональность SMTP. Следовательно, для клиентов с ограниченными возможностями **недопустимо** предполагать, что любой SMTP-сервер в Internet можно использовать для обработки (трансляции) почты. Если команда RCPT принята без предшествующей команды MAIL, сервер **должен** возвращать отклик 503 Bad sequence of commands[1]. Дополнительные параметры <rcpt-parameters> связываются с согласованным расширением сервиса SMTP (см. параграф 2.2).

[1]Недопустимый порядок следования команд.

Поскольку такая ошибка встречается достаточно часто, подчеркнём, что не допускается включение пробелов с любой стороны от знака двоеточия (:) после FROM в команде MAIL или после TO в команде RCPT. Точный синтаксис показан выше.

Третьим этапом транзакции является команда DATA (или соответствующая команда протокольного расширения).

```
DATA <CRLF>
```

Восприняв команду, сервер SMTP возвращает промежуточный отклик 354 Intermediate и рассматривает все последующие строки, вплоть (но не включая) до индикатора завершения почтовых данных, как текст сообщения. При успешном приёме всего текста сервер сохраняет полученные данные и возвращает отправителю отклик 250 OK.

Поскольку почтовые данные передаются через коммуникационный канал, завершение данных должно быть указано таким образом, чтобы можно было возобновить командный диалог. Протокол SMTP использует для обозначения конца почтовых данных точку в пустой строке. Для предотвращения ошибок при наличии такой последовательности в пользовательских данных применяется специальная процедура (transparency), описанная в параграфе 4.5.2.

Индикатор завершения почтовых данных также подтверждает почтовую транзакцию и говорит серверу SMTP, что нужно обрабатывать сохранённые пользовательские и почтовые данные. Восприняв данные, сервер SMTP возвращает отклик 250 OK. Сбой при обработке команды DATA может происходить только на двух этапах обмена данными.

Если команды MAIL и RCPT не были введены или были отвергнуты, сервер **может** возвращать отклик command out of sequence (503) или no valid recipients (554 – нет корректных получателей) в ответ на команду DATA. При получении одного из таких откликов (или любого отклика 5yz) для клиента **недопустима** передача данных серверу (точнее, передача данных **недопустима**, пока не будет получен отклик 354).

Если команда воспринята и передан отклик 354, невыполнение команды DATA может быть связано только с неполнотой почтовой транзакции (например, не указан адресат), недоступностью ресурсов (включая и неожиданную недоступность сервера) или отказом сервера от обработки сообщения в соответствии с заданной политикой или по иным причинам.

Однако на практике некоторые серверы не проверяют адресата после приёма текста сообщения. Таким серверам **следует** трактовать отказ для одного или нескольких получателей как «отказ обусловленный другим отказом» (subsequent failure) и возвращать почтовое сообщение, как указано в главе 6 и, в частности, в параграфе 6.1. Использование отклика 550 mailbox not found (или его эквивалента) после восприятия данных делает для клиента сложной или невозможной диагностику причины отказа.

При использовании формата RFC 822 ([28], [4]) почтовые данные включают элементы заголовка, такие как Date, Subject, To, Cc, From[1]. Серверам SMTP **не рекомендуется** отвергать сообщения на основе дефектов в заголовках RFC 822 и MIME (RFC 2045 [21]) или в теле сообщения. В частности, **недопустимо** отвергать сообщения, в которых число полей Resent не соответствует или Resent-to появляется без Resent-from и/или Resent-date.

Команды почтовых транзакций **должны** использоваться в приведённом выше порядке.

3.4. Пересылка для коррекции и обновления адресов

Поддержка пересылки чаще всего требуется для консолидации адресов и упрощения адресации в сети предприятия (или применительно к такой сети) и реже для случаев изменения адресов. Пересылка без уведомления отправителя (Silent forwarding) в целях обеспечения безопасности или сокрытия внутренней структуры весьма распространена сегодня в Internet.

В обоих перечисленных случаях приходится решать вопрос сокрытия (в некоторых случаях – безопасности) информации – следует ли показывать отправителю данные о пересылке почты. Это может быть особо важным, когда конечный адресат просто недоступен для отправителя. Следовательно, механизм пересылки, описанный в параграфе 3.2 RFC 821 и особенно строки откликов 251 (скорректированный получатель) и 551 на команду RCPT должны осторожно оцениваться при разработке и, когда это возможно, при настройке конфигурации системы (см. также параграф 7.4).

В частности:

- Сервер **может** пересылать сообщения, когда ему известно об изменении адреса. При такой пересылке сервер может предоставлять сведения о смене адреса с кодом 251 или «по-тихомуу» пересылать сообщение, возвращая код 250. При использовании кода 251 **недопустимо** предполагать, что клиент будет обновлять информацию об адресе получателя на основе принятого от сервера отклика.

Или:

- Сервер **может** отвергнуть или «завернуть» сообщения, когда их невозможно доставить по указанному адресу. В таких случаях сервер **может** сообщить о смене адреса в отклике 551 или отвергнуть сообщение как недоставляемое с кодом 550 без дополнительных сведений. При использовании кода 551 **недопустимо** предполагать, что отправитель будет обновлять адрес на основе полученных сведений или доводить эту информацию до пользователя.

Реализациям серверов SMTP, поддерживающим отклики с кодами 251 и/или 551, **следует** обеспечивать конфигурационный механизм, позволяющий отключить или ограничить дополнительную информацию для сайтов, которые могут использовать её нежелательным способом.

3.5. Команды для отлаживания адресов

3.5.1. Обзор

Протокол SMTP обеспечивает команды для проверки имён пользователей или получения содержимого списков рассылок. Такие операции осуществляются с помощью команд VRFY и EXPN, которые получают текстовые строки в

[1]Дата, тема, кому, копия, от кого.

качестве аргументов. Реализациям **следует** поддерживать команды VRFY и EXPN (особенности использования этих команд рассмотрены в параграфах 3.5.2 и 7.3).

Для команды VRFY параметром является имя пользователя, к которому может добавляться доменное имя (см. ниже). При получении нормального отклика (код 250) такой отклик **может** включать полное имя пользователя и **должен** включать название почтового ящика. Текст отклика **должен** использовать одну из двух возможных форм:

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

Когда имя, указанное в команде VRFY, может идентифицировать более одного почтового ящика, сервер **может** отметить неоднозначность или предложить в отклике несколько вариантов. Иными словами, в таких случаях возможен любой из перечисленных ниже вариантов отклика на команду VRFY:

```
553 User ambiguous
```

или

```
553- Ambiguous; Possibilities are
553-Joe Smith <jsmith@foo.com>
553-Harry Smith <hsmith@foo.com>
553 Melvin Smith <dweep@foo.com>
```

или

```
553-Ambiguous; Possibilities
553- <jsmith@foo.com>
553- <hsmith@foo.com>
553 <dweep@foo.com>
```

При нормальных условиях предполагается, что клиент, получивший отклик 553, доведёт эту информацию до пользователя. Использование приведённых здесь форм и ключевых слов user ambiguous (пользователя не определить однозначно) или ambiguous (неоднозначность), возможно дополненных расширенными кодами отклика (типа рассмотренных в RFC 3463 [25]), помогает при необходимости обеспечивать автоматический перевод на другие языки. Клиенты с высоким уровнем автоматизации и поддержкой других языков могут попытаться перевести отклик, возвратить пользователю нестандартную индикацию или предпринять некоторые автоматические операции типа обращения к службе каталогов для получения дополнительных данных перед возвратом отклика пользователю.

Для команды EXPN строка параметров идентифицирует список рассылки и при успешном выполнении команды возвращается отклик 250, который **может** включать полные имена пользователей и **должен** включать имена почтовых ящиков из списка.

На некоторых хостах различия между списками рассылок и псевдонимами выражены весьма слабо, поскольку оба типа записей могут сохраняться в единой структуре данных и возможны списки рассылок, содержащие единственный адрес. Если даётся запрос на применение команды VRFY к списку рассылок, позитивный отклик **может** быть возвращён, если направленное по адресу списка сообщение может быть доставлено кому-либо из списка, в остальных случаях **следует** возвращать сообщение об ошибке (например, отклик 550 That is a mailing list, not a User[1] или 252 Unable to verify members of mailing list[2]). Если делается запрос имени пользователя из списка, сервер **может** давать позитивный отклик, содержащий список из одного имени, или сообщение об ошибке (например, 550 That is a user name, not a mailing list[3]).

При успешном выполнении возвращаемый многострочный отклик (обычный для EXPN) содержит имя одного почтового ящика в каждой строке. Ситуации с неоднозначными запросами были рассмотрены выше.

Термин User name (имя пользователя) является недостаточно чётким и должен использоваться осмотрительно. Реализации команд VRFY и EXPN **должны**, по крайней мере, распознавать локальные почтовые ящики, как имена пользователей. Однако в сети Internet зачастую один хост обслуживает почту для множества доменов и хостам (особенно тем, которые работают с разными доменами) **следует** обеспечивать такую функциональность и воспринимать форму local-part@domain, как имя пользователя; хосты также **могут** распознавать, как имена пользователей, строки других типов.

Случай получения имён почтовых ящиков из списка рассылок требует многострочных откликов типа приведённого ниже (C – клиент, S – сервер; *прим. перев.*):

```
C: EXPN Example-People
S: 250-Jon Postel <Postel@isi.edu>
S: 250-Fred Fonebone <Fonebone@physics.foo-u.edu>
S: 250 Sam Q. Smith <SQSmith@specific.generic.com>
```

или

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

Символьная строка аргументов VRFY и EXPN не может быть дополнительно ограничена вследствие различных концепций именования пользователей и почтовых ящиков в разных реализациях. В некоторых системах аргументом команды EXPN может быть имя файла, содержащего список рассылки, но здесь опять приходится сталкиваться с различными соглашениями по именованию файлов в Internet. Отметим также, что в силу исторических причин вариации возвращаемых этими командами откликов достаточно велики, поэтому интерпретировать отклики **следует** очень осторожно и использовать только в целях диагностики.

3.5.2. Нормальные отклики VRFY

Когда возвращается нормальный код (2yz или 551) в результате запроса VRFY или EXPN, отклик **должен** включать имя почтового ящика в формате <local-part@domain>, где domain является полным доменным именем (FQDN). В ситуациях, исключающих нарушение требований данной спецификации, **может** возвращаться текстовая строка произвольной формы. Для облегчения анализа и разделения имени почтового ящика и данных человека (или компании) адрес **следует** выводить в угловых скобках. При возврате адресов (а не произвольной текстовой строки) команды EXPN и

[1]Это список рассылки, а не пользователь.
[2]Невозможно проверить членов списка.
[3]Это имя пользователя, а не список рассылки.

VRFY **должны** возвращать только корректные значения доменной части адреса, которые можно использовать в команде RCPT. Следовательно, если адрес может передаваться программе или другой системе, **должно** указываться имя почтового ящика, используемого для доступа к адресату. Возврат путей (явные маршруты source route) для команд VRFY и EXPN **недопустим**.

Реализациям серверов **следует** поддерживать обе команды VRFY и EXPN. В целях безопасности **может** обеспечиваться локальная возможность отключить любую из этих команд (или обе) с помощью конфигурационных параметров. Когда эти команды поддерживаются, не требуется обеспечивать их работу через трансляторы, если трансляция разрешена. Обе эти команды были **необязательными** в спецификации RFC 821, но команда VRFY стала обязательной в RFC 1123 [3]. Если команда EXPN поддерживается, она **должна** быть указана как расширение сервиса в отклике EHLO. VRFY **можно** указывать удобным способом, но в силу обязательности её поддержки, клиенты SMTP не обязаны проверять наличие этой команды в списке расширений перед началом её использования.

3.5.3. Значения откликов при успешном завершении VRFY или EXPN

Для серверов **недопустим** возврат откликов 250 на команды VRFY и EXPN, пока адрес реально не проверен. В частности, для сервера **недопустимо** возвращать код 250, если его действия ограничились проверкой корректности синтаксиса. В таких случаях **следует** возвращать код 502 (команда не реализована) или 500 (синтаксическая ошибка, команда не распознана). Как было указано, реализация (в смысле проверки адресов и возврата информации) команд VRFY и EXPN настоятельно рекомендуется. Следовательно, реализации, возвращающие код 500 или 502 для команды VRFY не являются полностью совместимыми с данной спецификацией.

Существуют ситуации, когда адрес представляется корректным, но не может быть проверен в реальном масштабе времени (в частности, когда сервер используется при обмене почтой для другого сервера или домена). «Видимая корректность» (Apparent validity) в таких случаях будет включать, по крайней мере, проверку синтаксиса и может также включать проверку возможности трансляции для указанного адреса. В таких случаях **следует** возвращать код 252. Эти ситуации связаны с вопросами проверки RCPT, рассмотренными в параграфе 2.1. Аналогично ситуации, описанной в 3.4, коды 251 и 551 могут использоваться для команд VRFY и EXPN, чтобы показать адреса, которые распознаны, но почта для них будет пересылаться или отвергаться. Реализациям в общем случае **следует** быть более жёсткими в вопросах проверки адресов для случая VRFY, нежели для команды RCPT, даже если это будет занимать немного больше времени.

3.5.4. Семантика и использование EXPN

Команда EXPN зачастую очень полезна для отладки и поиска проблем, связанных со списками рассылок и псевдонимами ко множеству адресов (multiple-target-address alias). Некоторые системы пытаются использовать поиск отправителя в списке рассылки для предотвращения дубликатов. Распространение системы псевдонимов с почтой в Internet для хостов (обычно записи MX и CNAME на серверах DNS), почтовых ящиков (различные типы локальных псевдонимов хоста) и в различных proxy-системах делает почти невозможной стратегию согласованного использования псевдонимов и почтовым системам **не следует** пытаться решить эту задачу.

3.6. Трансляция и маршрутизация почты

3.6.1. Маршрутизация, заданная отправителем, и трансляция

В общем случае доступность записей MX в DNS (RFC 1035 [2], RFC 974 [12]) избавляет от необходимости использования явно заданных маршрутов в почтовой системе Internet. С явной маршрутизацией почты связано множество проблем, делающих такое использование совершенно нежелательным. Клиентам SMTP **не следует** генерировать явные маршруты source route за исключением особых ситуаций. Серверы SMTP **могут** отказывать в трансляции или не воспринимать сообщения с указанным отправителем маршрутом. Обнаружив маршрутную информацию, сервер SMTP может игнорировать её и просто переслать почту конечному адресату, указанному в последнем элементе заданного маршрута, - серверам **следует** поступать именно так. Встречаются случаи некорректного использования имён адресатов, отсутствующих в записях DNS, с использованием преобразования имён на промежуточных хостах, указанных в маршруте source route. При исключении (игнорировании) заданного отправителем маршрута в таких случаях возникают проблемы. Это одна из нескольких причин, по которым для клиентов SMTP **недопустима** генерация некорректных маршрутов source route или путей, зависящих от последовательного преобразования имён.

Когда заданные отправителем маршруты не используются, процесс, описанный в RFC 821 для конструирования обратного пути из прямого, неприменим и обратный путь во время доставки будет просто адресом, указанным в команде MAIL.

3.6.2. Записи MX и трансляция

Транслирующий сервер SMTP обычно определяется из записи MX и не является системой окончательной доставки почты. Такой сервер может принимать или отвергать трансляцию почты аналогично восприятию или отказу для почты локальных пользователей. Если сервер принял трансляцию, от становится клиентом SMTP, организуя канал передачи следующему серверу SMTP, указанному в DNS (в соответствии с правилами, описанными в разделе 5), и передаёт ему почту. Если сервер отвергает трансляцию почты для какого-либо адреса, ему **следует** возвращать отклик 550.

В данной спецификации не рассматриваются вопросы верификации путей возврата для передачи уведомлений о доставке. В последнее время были выполнены работы (такие, как SPF [29] и DKIM [30] [31]) по обеспечению способов проверки корректности адресов возврата и их принадлежности лицу, действительно отправившему сообщение. Сервер **может** попытаться проверить путь возврата перед использованием этого адреса для передачи уведомления о доставке, однако здесь не определяются способы такой проверки и не даётся рекомендация по выбору способа проверки.

3.6.3. Серверы представления сообщений как трансляторы

Существует множество клиентов, передающих почту (часто эти же программы служат для приёма почты по протоколу POP3 или IMAP), которые не обеспечивают полную поддержку данной спецификации (например, поддержка очередей

для последующей передачи). Для таких клиентов обычной практикой является организация частного соглашения с сервером для отправки ему всей почты с целью последующей обработки и доставки. Как указано здесь, SMTP не является идеальным решением для таких задач. Разработан стандартизованный протокол представления почтовых сообщений (RFC 4409 [18]), учитывающий опыт использования систем SMTP. В любом случае, частный характер соглашения между сервером и клиентами выводит этот вопрос за пределы данной спецификации.

Важно отметить, что записи MX могут указывать на серверы SMTP, которые действуют как шлюзы в другие среды, а не только выполняют трансляцию и окончательный приём почты (см. параграф 3.8 и раздел 5).

Если сервер SMTP принял на себя задачу трансляции почты и позднее обнаружил, что получатель указан некорректно или почту невозможно доставить по тем или иным причинам, этот сервер **должен** создать уведомление о невозможности доставки почты и переслать его отправителю недоставленного сообщения, указанному в обратном пути. Для уведомления **следует** (по возможности) использовать стандартные форматы (см., например RFC 3461 [32] и RFC 3464 [33]).

Это уведомление должно передаваться сервером SMTP с хоста-транслятора или хоста, который обнаружил невозможность доставки. Естественно, что для серверов SMTP **недопустима** отправка уведомлений о невозможности доставки уведомлений[1]. Одним из способов предотвращения петель при передаче сообщений об ошибках является использование пустой строки обратного пути в команде MAIL при передаче уведомления. При передаче такого сообщения строка обратного пути **должна** быть пустой – null (см. параграф 4.5.5). Команда MAIL с пустым обратным путём имеет вид:

```
MAIL FROM:<>
```

Как указано в параграфе 6.4, транслятору SMTP не нужно проверять и обрабатывать заголовки и тело транслируемых сообщений, а также **недопустимо** предпринимать какие-либо любые действия по отношению к сообщению, за исключением добавления к заголовку строки Received: (см. параграф 4.4) и (необязательной) попытки обнаружения петель в почтовой системе (см. параграф 6.3). Естественно, запрет распространяется и на изменение любых полей заголовка и текста сообщения (см. также параграф 7.9).

3.7. Почтовые шлюзы

Описанные выше трансляторы работают в транспортной среде Internet SMTP, однако записи MX и разные формы явной маршрутизации могут потребовать использования промежуточных серверов SMTP, которые будут обеспечивать преобразование почты между различными транспортными системами. Как было отмечено в параграфе 2.3.10, такие системы, работающие на границах между двумя системами транспортного сервиса, называются шлюзами или почтовыми шлюзами[2].

Шлюзование почты между различными почтовыми средами (разные форматы и протоколы) является сложной задачей, стандартизация которой также непроста. Однако можно сформулировать некие требования общего плана для шлюзов между Internet и другими почтовыми средами.

3.7.1. Поля заголовка при использовании шлюзов

Поля заголовка могут быть при необходимости переписаны, когда сообщение передаётся через границу между почтовыми средами. Несмотря на приведённые в параграфе 6.4 запреты, локальная часть адреса получателя может быть изменена шлюзом; допускается также проверка содержимого почты.

Другие почтовые системы при передаче сообщений в Internet часто используют подмножество заголовков RFC 822 или обеспечивает похожую функциональность с использованием другого синтаксиса, но некоторые из таких почтовых систем не имеют эквивалента конвертов SMTP. Следовательно, когда сообщение покидает почтовую среду Internet, может потребоваться включение информации из конверта SMTP в заголовок сообщения. Возможным решением будет создание новых полей заголовка для передачи информации из конверта (например, X-SMTP-MAIL: и X-SMTP-RCPT:). Однако такое решение потребует изменения почтовых программ в чужой среде и может привести к разглашению частной информации (см. параграф 7.2).

3.7.2. Строки Received при использовании шлюзов

При пересылке сообщения в среду Internet или из неё шлюз **должен** включить в заголовок свою строку Received:, но **недопустимо** менять строки Received, уже имеющиеся в заголовке.

Поля Received: сообщений из чужих сред могут не соответствовать данной спецификации. Однако наиболее важным аспектом использования строк Received: является диагностика сбоев в почтовой системе и такая отладка может быть сильно осложнена шлюзами, которые пытаются «исправить» строки Received:. Другим важным аспектом обработки трассировочных полей из других (не SMTP) сред является то, что для принимающей системы **недопустимо** отвергать почту на основе формата полей трассировки и **следует** сохранять максимум здравомыслия при встрече с неожиданной информацией или форматами полей трассировки.

Шлюзу **следует** указывать среду и протокол в поле via строки Received, создаваемый шлюзом.

3.7.3. Адресация при использовании шлюзов

Со стороны Internet шлюзу **следует** воспринимать все корректные форматы адресов в командах SMTP и заголовках RFC 822, а также все корректные сообщения RFC 822. Генерируемые шлюзом адреса и заголовки **должны** соответствовать применимым стандартам Internet (включая данную спецификацию и RFC 5322 [4]). Шлюзы подчиняются тем же правилам обработки маршрутов source route, которые описаны в параграфе 3.3 для других систем SMTP.

[1] О невозможности доставки почты. *Прим. перев.*
[2] Gateway, gateway SMTP.

3.7.4. Другие поля заголовков при использовании шлюзов

Шлюз **должен** обеспечивать соответствие требованиям Internet всех полей заголовков в сообщениях, передаваемых в почтовую среду Internet. В частности, все адреса в полях From:, To:, Cc: и т. п. **должны** преобразовываться (если нужно) в соответствии с синтаксисом RFC 5322 [4], **должны** указывать только полные доменные имена и **должны** быть эффективны и полезны для передачи откликов. Алгоритму, используемому для преобразования почты Internet в другие форматы, **следует** обеспечивать доставку сообщений об ошибках из чужой почтовой среды по пути возврата в конверте SMTP, а не отправителю, указанному в поле From:, Sender: (или других полях) заголовка сообщения.

3.7.5. Конверты при работе со шлюзами

При пересылке сообщений из других сред в Internet шлюзу **следует** устанавливать в конверте путь возврата в соответствии с адресом возврата сообщений об ошибках, если этот адрес предоставляется чужой средой. Если в чужой среде нет эквивалентной концепции, шлюз должен выбрать и использовать наилучшее приближение (адрес исходного отправителя сообщения при отсутствии других вариантов).

3.8. Прерывание сеансов и соединений

Соединение SMTP разрывается при получении от клиента команды QUIT. Сервер возвращает в ответ на эту команду позитивный отклик и закрывает соединение.

Для серверов SMTP **недопустимо** преднамеренно закрывать соединения в нормальных условиях (см. параграф 7.8) за исключением следующих ситуаций:

- После получения команды QUIT и отклика на неё с кодом 221.
- После определения необходимости отключения (shut down) сервиса SMTP и возврата кода 421. Такой отклик может выдаваться после получения сервером любой команды или (при необходимости) асинхронно (независимо от команд) в предположении, что клиент будет получать отклик после ввода следующей команды.
- После тайм-аута (см. параграф 4.5.3.2) в процессе ожидания от клиента команды или данных.

В частности, разрыв соединения сервером в ответ на непонятную команду является нарушением данной спецификации. Ожидается, что серверы будут терпимы к неизвестным командам, возвращая в ответ на них код 500 и ожидая дальнейших инструкций от клиента.

Серверам SMTP, которые отключаются в результате внешнего воздействия, **следует** пытаться передать клиенту строку, содержащую код 421, до завершения работы. Клиент SMTP обычно будет получать код 421 после передачи следующей команды.

Клиентам SMTP, узнавшим о закрытии соединения, сбросе или других коммуникационных сбоях вследствие неконтролируемых клиентом событий (в нарушение данной спецификации, иногда неизбежное), для обеспечения устойчивости почтовой системы **следует** трактовать почтовую транзакцию, как при получении отклика 451 и действовать в соответствии с этим.

3.9. Почтовые списки и псевдонимы

Хостам SMTP **следует** поддерживать как псевдонимы, так и списки для преобразования адресов при групповой рассылке сообщений. Когда сообщение доставляется или пересылается по каждому адресу из списка, адрес возврата в конверте (MAIL FROM:) **должен** заменяться на адрес администратора списка. Однако в таких случаях заголовок сообщения (RFC 5322 [4]) **должен** сохраняться неизменным; в частности, не должно меняться поле From.

Одним из важных свойств почтовой системы является механизм доставки одного сообщения множеству адресатов за счёт преобразования (expanding или exploding) псевдоадреса в список реальных адресов получателей. Когда сообщение направляется по такому псевдоадресу (иногда его называют exploder), копия этого сообщения пересылается по каждому адресу из списка. Серверу **следует** просто использовать адреса из списка, применение эвристики или проверки соответствия для исключения некоторых адресов (например, отправителя исходного сообщения) настоятельно не рекомендуется. Псевдоадреса называют списками (list, mail list) или псевдонимами (alias) в зависимости от способа получения адресов из списка.

3.9.1. Псевдонимы

Для преобразования псевдонима почтовая программа-получатель просто заменяет в заголовке псевдоадрес преобразованным адресом из псевдонима, сохраняя неизменными остальную часть конверта и тело сообщения. После этого сообщения доставляются или пересылаются по всем адресам.

3.9.2. Списки

Почтовые списки обеспечивают перераспределение (redistribution), а не пересылку (forwarding) сообщений. Для преобразования списка почтовая программа-получатель заменяет в конверте псевдоадрес реальными адресами из списка. Адрес возврата в конверте заменяется так, чтобы все сообщения об ошибках приходили по адресу администратора списка (не отправителя сообщения), который обычно контролирует содержимое списков и доставку. Отметим, что основное различие между обработкой псевдонимов (параграф 3.9.1) и списков (данный параграф) заключается в изменении адреса возврата при рассылке по списку. Хотя списки ограничивают набор операций обработки описанными здесь действиями, они являются попыткой эмуляции АДП (MTA); такие списки можно рассматривать как продолжение процедур транзита почты.

Существуют списки, которые поддерживают дополнительные (иногда значительные) изменения конвертов и тела сообщений. Такие списки необходимо рассматривать как полные ППА (MUA), которые принимают сообщение и передают новое.

4. Спецификации SMTP

4.1. Команды SMTP

4.1.1. Синтаксис и семантика команд

Команды SMTP определяют передачу почты и функции почтовой системы, запрашиваемые пользователем. Команды представляют собой текстовые строки, завершающиеся последовательностью <CRLF>. Команда, как таковая, представляет собой строку букв, завершаемую пробелом <SP> (при наличии параметров) или <CRLF>. В целях повышения уровня взаимодействия получателям SMTP следует быть терпимыми к пробелам перед завершающей последовательностью <CRLF>. Синтаксис локальной части имени почтового ящика соответствует соглашениям принимающего сайта и синтаксису, описанному в параграфе 4.1.2. Команды SMTP обсуждаются ниже, а рассмотрению откликов посвящён параграф 4.2.

Почтовая транзакция включает несколько объектов данных, используемых в качестве аргументов различных команд. Обратный путь является аргументом команды MAIL, прямой путь – аргументом RCPT, а почтовые данные – аргументом команды DATA. Эти аргументы или объекты данных должны передаваться и сохраняться до завершения почтовой транзакции. Для каждого типа данных (прямой и обратный путь и почтовые данные) используются различные буферы (буферы прямых и обратных путей, буфер данных). Конкретная команда приводит к добавлению информации в конец соответствующего буфера или созданию одного или нескольких новых буферов.

Некоторые команды (RSET, DATA, QUIT) не поддерживают параметров. В отсутствие специфических расширений, предлагаемых сервером и принимаемых клиентом, для последних **недопустимо** передавать параметры таким командам, а серверу **следует** отвергать команды, как в случае некорректного синтаксиса.

4.1.1.1. Расширенное (EHLO) или стандартное (HELO) приветствие

Эти команды используются для представления SMTP-клиента серверу SMTP. Поле аргументов содержит полное доменное имя клиента SMTP, если такое имя доступно. В тех случаях, когда клиент SMTP не имеет значимого доменного имени (например, при динамическом выделении адресов и недоступности обратного преобразования), клиентам **следует** передавать полный адрес (см. параграф 4.1.3).

В RFC 2821 и неформальной практике прошлых лет рекомендуется сопровождать точный адрес информацией, которая поможет идентифицировать клиентскую систему. Такая практика не получила широкого распространения и многие серверы SMTP рассматривают дополнительную информацию, как ошибку. В целях обеспечения взаимодействия серверам полезно принимать такую информацию, но клиентам SMTP не следует передавать её.

Сервер SMTP представляет себя клиенту в данном соединении с помощью отклика на команду приветствия.

Клиентам SMTP **следует** начинать сессию SMTP с помощью команды EHLO. Если сервер SMTP поддерживает расширенные службы SMTP, он будет передавать позитивный отклик, сообщение об отказе или сообщение об ошибке. Если сервер SMTP (в нарушение данной спецификации) не поддерживает никаких расширений SMTP, он будет генерировать сообщение об ошибке. Старые клиенты SMTP **могут** (как обсуждалось выше) использовать команду HELO (в соответствии с RFC 821) взамен EHLO, а серверы **должны** поддерживать команды HELO и давать на них правильный отклик. В любом случае клиент **должен** использовать команду HELO или EHLO до начала почтовой транзакции.

Эти команды и отклики 250 OK в ответ на них подтверждают, что клиент и сервер SMTP находятся в начальной стадии, в которой нет выполняемых транзакций, а все таблицы состояния и буфера ещё пустые.

Синтаксис:

```
ehlo            = "EHLO" SP ( Domain / address-literal ) CRLF
helo            = "HELO" SP Domain CRLF
```

Обычно в ответ на команду EHLO возвращается многострочный отклик, каждая строка которого содержит ключевое слово и может включать один или несколько параметров. В соответствии с требованиями к нормальному синтаксису многострочных откликов ключевые слова следуют после кода 250 и дефиса (для всех строк, кроме последней) или пробела (в последней строке). Ниже приведён пример позитивного отклика с использованием нотации ABNF и символов завершения из RFC 5234 [7]:

```
ehlo-ok-rsp     = ( "250" SP Domain [ SP ehlo-greet ] CRLF )
                  / ( "250-" Domain [ SP ehlo-greet ] CRLF
                  *( "250-" ehlo-line CRLF )
                  "250" SP ehlo-line CRLF )
ehlo-greet      = 1*(%d0-9 / %d11-12 / %d14-127)
                  ; строка любых символов, кроме CR и LF

ehlo-line       = ehlo-keyword *( SP ehlo-param )

ehlo-keyword    = (ALPHA / DIGIT) *(ALPHA / DIGIT / "-")
                  ; дополнительный синтаксис ehlo-params зависит от ehlo-keyword

ehlo-param      = 1*(%d33-126)
                  ; любые символы, включая <SP> и все коды управления (US-ASCII 0-31 и 127,
                  ; включительно)
```

Хотя в команде EHLO можно использовать любую комбинацию строчных и прописных букв, команда всегда **должна** распознаваться и обрабатываться как EHLO (заглавные буквы) – это просто расширение практики, указанной в RFC 821 и параграфе 2.4.

Отклик на команду EHLO **должен** включать ключевые слова (и связанные с ними параметры при наличии последних) для всех команд, не перечисленных как «обязательные» в параграфе 4.5.1, за исключением команд для приватного использования, описанных в параграфе 4.1.5. Команды для приватного использования также **можно** включать в список.

4.1.1.2. Начало транзакции (MAIL)

Эта команда служит для инициирования почтовой транзакции, в которой почтовые данные доставляются на сервер SMTP, который, в свою очередь, доставляет почту в один или несколько почтовых ящиков или передаёт её другой почтовой системе (возможно, с использованием SMTP). Поле аргументов содержит обратный путь и может включать дополнительные параметры. В общем случае команда MAIL может передаваться только при отсутствии незавершённых почтовых транзакций (см. параграф 4.1.4).

Обратный путь указывает почтовый ящик отправителя. В силу исторических причин почтовому ящику может предшествовать список хостов, но такая практика в настоящее время осуждается (см. Приложение C). В некоторых типах сообщений-отчетов, отклики на которые могут порождать петли (например, уведомления о доставке или невозможности доставки) поле обратного пути является пустым (см. параграф 3.6).

Эта команда очищает буферы обратного пути, прямого пути и почтовых данных, а также помещает информацию из строки параметров в буфер обратного пути.

Если согласовано использование расширенного сервиса, команда MAIL может содержать дополнительные параметры.

Синтаксис:

```
mail = "MAIL FROM:" Reverse-path [SP Mail-parameters] CRLF
```

4.1.1.3. Получатель (RCPT)

Эта команда служит для идентификации отдельного получателя почтовых данных; при необходимости задать множество получателей команда повторяется соответствующее число раз. Поле аргументов содержит прямой путь и может включать дополнительные параметры.

Прямой путь обычно указывает почтовый ящик получателя. Передающим системам **не следует** генерировать дополнительный список хостов, известный как source route (маршрут, заданный отправителем). Принимающие системы **должны** распознавать синтаксис source route, но им **следует** вырезать спецификацию этого маршрута, используя взамен доменное имя, связанное с почтовым ящиком, как будто отправитель вообще не задавал маршрута.

Подобно этому, трансляторам **следует** пропускать или игнорировать source route, а имена **недопустимо** копировать в поле обратного пути. Когда почта приходит к конечному адресату (прямой путь содержит только почтовый ящик получателя), сервер SMTP помещает сообщение в почтовый ящик адресата в соответствии с принятыми соглашениями.

Эта команда добавляет свой аргумент forward-path в буфер прямого пути, не меняя содержимого буферов обратного пути и почтовых данных.

Например, почта, полученная транслятором xyz.com и содержащая в конверте команды:

```
MAIL FROM:<userx@y.foo.org>
RCPT TO:<@hosta.int,@jkl.org:userc@d.bar.org>
```

обычно будет пересылаться непосредственно на хост d.bar.org с командами в конверте**:**

```
MAIL FROM:<userx@y.foo.org>
RCPT TO:<userc@d.bar.org>
```

Как указано в Приложении C, хост xyz.com **может** также транслировать почту через другой хост, используя в конверте команды:

```
MAIL FROM:<userx@y.foo.org>
RCPT TO:<@hosta.int,@jkl.org:userc@d.bar.org>
```

или (для трансляции через jkl.org):

```
MAIL FROM:<userx@y.foo.org>
RCPT TO:<@jkl.org:userc@d.bar.org>
```

Попытки такого использования трансляторов в настоящее время осуждаются. Поскольку хосты не обязаны транслировать почту, xyz.com **может** отвергнуть сообщение при получении команды RCPT, используя отклик 550 (отказ в соответствии с используемыми правилами).

Если согласовано использование расширенного сервиса, команда RCPT может также включать параметры, связанные с конкретным типом расширения, предлагаемого сервером. Для клиента **недопустима** передача параметров, кроме тех, которые связаны с расширением, предложенным сервером в отклике EHLO.

Синтаксис:

```
rcpt = "RCPT TO:" ( "<Postmaster@" Domain ">" / "<Postmaster>"
       / Forward-path ) [SP Rcpt-parameters] CRLF
```

Отметим, что в отличие от обычных правил для локальных путей регистр символов в строке Postmaster принимается во внимание.

4.1.1.4. Данные (DATA)

Получатель обычно возвращает отклик 354 на команду DATA и после этого трактует дальнейшие строки (символьные последовательности, завершающиеся <CRLF>, как сказано в параграфе 2.3.7), как почтовые данные от отправителя. Эта команда добавляет почтовые данные в конец буфера данных. Данные могут включать любые из 128 символов ASCII, хотя опыт показывает, что использование управляющих символов (кроме SP, HT, CR, LF) может вызывать проблемы, поэтому **следует** избегать таких символов.

Данные завершаются строкой, содержащей только точку и последовательность завершения строки (в потоке символов это будет <CRLF>.<CRLF>, см. параграф 4.5.2). Такая последовательность символов указывает на завершение потока данных. Первая последовательность <CRLF> на самом деле завершает последнюю строку почтовых данных (текста сообщения) или (при отсутствии данных) – командную строку DATA (случай отсутствия данных не соответствует этой спецификации, поскольку он требует, чтобы не передавалось ни трассировочных полей заголовка, ни заголовка сообщения, требуемых RFC 5322 [4]). **Недопустимо** добавление лишних последовательностей <CRLF>, поскольку это будет приводить к вставке пустой строки в сообщение. Единственным исключением из этого правила является

обработка сообщений, переданных исходному отправителю без завершающей последовательности <CRLF> в последней строке; в таких случаях отправляющая сообщение система SMTP **должна** отвергнуть сообщение как некорректное или добавить <CRLF> в конце, чтобы принимающий сервер SMTP смог зафиксировать условие end of data (конец сообщения).

Использование строк, завершающихся одиночным символом <LF>[1], как это принято в некоторых UNIX-системах, порождает значительно больше проблем, нежели решает и для серверов SMTP такой подход **недопустим**, даже во имя повышения отказоустойчивости. В частности, последовательности <LF>.<LF> **недопустимо** трактовать как эквивалент последовательности <CRLF>.<CRLF> для завершения почтовых данных.

Получение индикатора завершения данных требует от сервера обработки сохранённых данных почтовой транзакции. При этой обработке используется содержимое буферов прямого и обратного пути, а также буфера данных. По завершении команды буферы очищаются. Если обработка команды завершилась успешно, получатель **должен** передать отклик OK, а при неудаче – отклик о неудачной попытке. Модель SMTP не допускает частичных отказов на этом этапе – сообщение или воспринимается сервером для доставки с возвратом позитивного отклика, или не принимается и сервер возвращает негативный отклик. После передачи позитивного отклика на завершение приёма данных сервер принимает на себя полную ответственность за это сообщение (см. параграф 6.1). При обнаружении ошибок впоследствии **должны** передаваться почтовые уведомления об ошибках, как сказано в параграфе 4.4.

Когда сервер SMTP воспринимает сообщение для трансляции или окончательной доставки, он помещает трассировочную запись, которую также называют time stamp line (строка с временной меткой) или Received в верхней части почтовых данных. Эта запись показывает хост, передавший сообщение, хост-приемник (сервер), а также дату и время приёма сообщения. Транслируемые сообщения могут содержать на финальном этапе множество трассировочных записей. Детальное описание трассировки и синтаксиса записей приводится в параграфе 4.4.

Дополнительную информацию по обработке команд DATA можно найти в параграфе 3.3.

Синтаксис:

```
data = "DATA" CRLF
```

4.1.1.5. Сброс (RSET)

Эта команда служит для прерывания текущей почтовой транзакции. Все сохранённые в буферах данные **должны** быть отброшены с очисткой буферов и таблиц состояния. Принимающая сторона в ответ на команду RSET **должна** передать отклик 250 OK без дополнительных аргументов. Команду RSET клиент может вводить в любой момент транзакции. Эта команда является эквивалентом NOOP (не выполняется никаких действий) при введении сразу после EHLO, до первого использования EHLO в данном сеансе, после завершения и подтверждения передачи данных или непосредственно перед командой QUIT. Для серверов SMTP **недопустимо** закрывать соединение в результате получения команды RSET – для разрыва соединения служит команда QUIT (см. параграф 4.1.1.10).

Поскольку обработка команд EHLO требует некоторых дополнительных операций на сервере, использование команды RSET обычно более эффективно, чем повторный ввод EHLO, хотя формальная семантика одинакова.

Существуют обстоятельства (не контролируемые данной спецификацией), при которых сервер SMTP может получить индикацию разрыва или сброса соединения на нижележащем уровне TCP. Для сохранения отказоустойчивости почтовых систем серверам SMTP **следует** быть готовыми к таким ситуациям и трактовать их как получение команды QUIT до потери соединения.

Синтаксис:

```
rset = "RSET" CRLF
```

4.1.1.6. Проверка (VRFY)

Эта команда просит получателя подтвердить аргументы, идентифицирующие пользователя или почтовый ящик. Если это имя пользователя, возвращается информация в соответствии с описанием в параграфе 3.5.

Данная команда не воздействует на содержимое буферов обратного и прямого пути, а также буфера почтовых данных.

Синтаксис:

```
vrfy = "VRFY" SP String CRLF
```

4.1.1.7. Преобразование списка (EXPN)

Эта команда просит подтвердить аргументы, идентифицирующие список рассылки, и (при наличии указанного списка) возвращает список членов. При успешном завершении команды возвращается информация, описанная в параграфе 3.5. Этот отклик будет содержать множество строк за исключением тривиальных случаев списка с одним адресом.

Данная команда не воздействует на содержимое буферов обратного и прямого пути, а также буфера данных.

Синтаксис:

```
expn = "EXPN" SP String CRLF
```

4.1.1.8. Справка (HELP)

По этой команде сервер возвращает краткие справочные сведения о командах и аргументах. Команда **может** использовать в качестве аргумента имя другой команды для получения соответствующей справки.

Данная команда не воздействует на содержимое буферов обратного и прямого пути, а также буфера данных. Команда может использоваться в любой момент.

Серверам **следует** поддерживать команду HELP без аргументов и **можно** поддерживать команду с аргументами.

Синтаксис:

```
help = "HELP" [ SP String ] CRLF
```

[1]Перевод строки без возврата каретки. *Прим. перев.*

4.1.1.9. Пустая операция (NOOP)

Эта команда не влияет на значения параметров и выполнение введённых ранее команд. По команде сервер просто передаёт отклик 250 OK.

Данная команда не воздействует на содержимое буферов обратного и прямого пути, а также буфера данных и может вводиться в любой момент. При наличии у команды параметров серверу **следует** игнорировать их.

Синтаксис:

```
noop = "NOOP" [ SP String ] CRLF
```

4.1.1.10. Завершение сеанса (QUIT)

Получив эту команду сервер **должен** возвратить отклик и 221 OK закрыть канал передачи.

Для получателя **недопустим** преднамеренный разрыв соединения до получения команды QUIT и отклика на неё (даже при возникновении ошибок). Для сервера **недопустим** преднамеренный разрыв соединения до передачи команды QUIT и **следует** дождаться отклика на неё (даже при возникновении ошибок в результате выполнения предыдущих команд). Если соединение закрыто преждевременно в нарушение сказанного выше или в результате системного или сетевого сбоя, сервер **должен** прервать все незавершённые транзакции, не отказываясь от выполненных транзакций, и (в общем случае) **должен** действовать, как при получении информации об ошибке во время выполнения команды или транзакции (т. е. отклик 4yz).

Команда QUIT может быть введена в любой момент. Незавершённая почтовая транзакция прерывается.

Синтаксис:

```
quit = "QUIT" CRLF
```

4.1.1.11. Mail-Parameter and Rcpt-Parameter Error Responses

Если сервер SMTP не распознает или не поддерживает один или более параметров конкретной команды MAIL FROM или RCPT TO, он будет возвращать код 555.

Если по той или иной причине сервер временно не способен принят один или множество параметров, связанных с командой MAIL FROM или RCPT TO, а определение соответствующего параметра не требует использования иного кода, серверу следует возвращать код 455.

Ошибки, связанные с конкретными параметрами и их значениями, определяются в соответствующих RFC.

4.1.2. Синтаксис аргументов команд

Ниже приведён синтаксис полей аргументов перечисленных выше команд (по возможности, используется синтаксис, описанный в RFC 5234 [7]). Некоторые из приведённых ниже вариантов используются только с маршрутами source route, как описано в Приложении C. Обозначения, не определённые здесь (типа ALPHA, DIGIT, SP, CR, LF, CRLF), описаны в разделе 6 RFC 5234 [7] или при формальном определении синтаксиса сообщений в RFC 5322 [4].

```
Reverse-path    = Path / "<>"
Forward-path    = Path
Path            = "<" [ A-d-l ":" ] Mailbox ">"
A-d-l           = At-domain *( "," At-domain )
                ; отметим, что эта форма, называемая source route, должна приниматься,
                ; её не следует генерировать и следует игнорировать.
At-domain       = "@" Domain
Mail-parameters  = esmtp-param *(SP esmtp-param)
Rcpt-parameters  = esmtp-param *(SP esmtp-param)
esmtp-param     = esmtp-keyword ["=" esmtp-value]
esmtp-keyword   = (ALPHA / DIGIT) *(ALPHA / DIGIT / "-")
esmtp-value     = 1*(%d33-60 / %d62-126)
                ; любые символы кроме =, SP и управляющих кодов. Если эта строка
                ; представляет собой почтовый адрес (например, Mailbox), следует
                ; использовать синтаксис xxtext [32].
Keyword         = Ldh-str
Argument        = Atom
Domain          = sub-domain *("." sub-domain)
sub-domain      = Let-dig [Ldh-str]
Let-dig         = ALPHA / DIGIT
Ldh-str         = *( ALPHA / DIGIT / "-" ) Let-dig
address-literal  = "[" ( IPv4-address-literal /
                  IPv6-address-literal /
                  General-address-literal ) "]"
                  ; см. параграф 4.1.3
Mailbox         = Local-part "@" ( Domain / address-literal )
Local-part      = Dot-string / Quoted-string
                ; регистр символов может различаться
Dot-string      = Atom *("."  Atom)
Atom            = 1*atext
Quoted-string   = DQUOTE *QcontentSMTP DQUOTE
QcontentSMTP    = qtextSMTP / quoted-pairSMTP
quoted-pairSMTP  = %d92 %d32-126
                  ; т. е., обратная дробная черта, за которой следует символ
                  ; псевдографики ASCII (включая \) или пробел (SP)
qtextSMTP       = %d32-33 / %d35-91 / %d93-126
                ; т. е., кавычках допускается использование любых символов псевдографики
              ; ASCII (за исключением \ и двойных кавычек) или пробелов без символа
                ; обратной дробной черты
String          = Atom / Quoted-string
```

Хотя в приведённом выше описании требования к локальной части адреса относительно либеральны, хостам, принимающим почту, **следует** избегать организации почтовых ящиков, для которых Local-part требует (или использует) форму Quoted-string или различается регистр символов. Для любых задач, требующих генерации или сравнения полей Local-part, все формы Quoted-string **должны** трактоваться как эквивалентные и передающим системам следует передавать форму, использующую минимальное квотирование.

Недопустимо определять почтовые ящики таким образом, чтобы в SMTP требовалось использование символов, не входящих в набор ASCII (октетов с 1 в старшем бите) или управляющих кодов ASCII (десятичные значения 0-31 и 127). Такие символы **недопустимо** использовать в командах MAIL и RCPT или других командах, содержащих имена почтовых ящиков.

Отметим, что обратный слэш (\) относится к символам квотирования, используемым для индикации буквального (literally) использования следующего символа (взамен обычной интерпретации). Например, запись "Joe\, Smith" соответствует “Joe, Smith”, т. е. Запятая после знака \ трактуется именно как запятая, а не специальный символ.

Для обеспечения взаимодействия и совместимости с DNS в именовании и приложениях (см., например, параграф 2.3.1 базового стандарта DNS - RFC1035 [2]) недопустимо включать в метки доменных имён для клиентов и серверов SMTP никакие символы, кроме букв латиницы, цифр и дефиса. В частности, символ подчёркивания (underscore) использовать нельзя. Серверы SMTP, получающие команды с некорректными символами (при отсутствии других причин для отказа), **должны** отвергать такие команды с возвратом отклика 501 (это правило, подобно другим, может быть изменено расширениями SMTP).

4.1.3. «Дословные» адреса

Иногда хост не знает доменного имени и почтовая связь (в частности, передача сообщений об ошибках) блокируется. Для решения этой проблемы в качестве альтернативы доменному имени может использоваться специальная форма адреса (literal address). Для адресов IPv4 эта форма использует десятичное представление байтов IP-адреса с разделением точками. Адреса заключаются в квадратные скобки (например, [123.255.37.2]), которые говорят об использовании адреса IPv4 в десятичном представлении с разделением точками. Для IPv6 и других форм адресации, которые могут быть в последствии стандартизованы, форма включает стандартизованный тег, идентифицирующий синтаксис адреса, (двоеточие - :) и собственно адрес в формате, заданном стандартом [например, RFC 4291 [8] для IPv6].

В частности, используются следующие варианты:

```
IPv4-address-literal  = Snum 3("."  Snum)
IPv6-address-literal  = "IPv6:" IPv6-addr
General-address-literal  = Standardized-tag ":" 1*dcontent
Standardized-tag  = Ldh-str
                  ; должен быть опубликован в RFC со статусом Standards-Track  и
                  ; зарегистрирован IANA
dcontent       = %d33-90 / ; Печатаемый символ US-ASCII
                 %d94-126    ; исключая "[", "\", "]"
Snum           = 1*3DIGIT
                 ; значения от 0 до 255 в десятичном представлении
IPv6-addr      = IPv6-full / IPv6-comp / IPv6v4-full / IPv6v4-comp
IPv6-hex       = 1*4HEXDIG
IPv6-full      = IPv6-hex 7(":" IPv6-hex)
IPv6-comp      = [IPv6-hex *5(":" IPv6-hex)] "::"
                 [IPv6-hex *5(":" IPv6-hex)]
                 ; "::" представляет по крайней мере две 16-битовых группы нулей. В
                 ; дополнение может использоваться до 6 групп.
IPv6v4-full    = IPv6-hex 5(":" IPv6-hex) ":" IPv4-address-literal
IPv6v4-comp    = [IPv6-hex *3(":" IPv6-hex)] "::" [IPv6-hex *3(":" IPv6-hex) ":"]
                 IPv4-address-literal
               ; :: представляет по крайней мере две 16-битовых последовательности нулей
               ; в дополнение к :: может присутствовать не более 4 групп и
               ; IPv4-address-literal
```

4.1.4. Порядок команд

Для порядка использования команд существуют некоторые ограничения.

Сеанс, который будет включать почтовую транзакцию, **должен** быть сначала инициализирован командой EHLO. Серверам SMTP **следует** воспринимать без инициализации команды, не использующие почтовых транзакций (например, VRFY или EXPN).

Команда EHLO **может** вводиться клиентом в действующем сеансе. При первом использовании команды в данной сессии сервер SMTP **должен** очистить все буферы и сбросить состояние, как при получении команды RSET. Иными словами, последовательность команд RSET - EHLO является избыточной, и мало полезна ввиду выполнения ненужных повторяющихся действий.

Если команда EHLO неприемлема для сервера SMTP, он **должен** возвращать отклик 501, 500, 502 или 550. Сервер SMTP **должен** сохранять после передачи таких откликов состояние, которое было до получения команды EHLO.

Клиент SMTP **должен** (по возможности) предоставлять в параметрах команд EHLO первичное доменное имя (не CNAME или MX) своего хоста, как сказано в параграфе 2.3.5. Если это невозможно (например, клиент использует динамический адрес и не имеет явного имени), **следует** взамен имени использовать «дословный» адрес.

Сервер SMTP **может** проверять соответствие доменного имени в команде EHLO реальному IP-адресу клиента. Однако для сервера **недопустимо** отвергать сообщение по результатам проверки. Эти результаты могут использоваться для протоколирования и трассировки. Отметим, что этот запрет применим только к соответствию параметра и адреса IP; дополнительное рассмотрения вопросов отказа от приёма входящих соединений или почтовых сообщений проводится в параграфе 7.9.

Команды NOOP, HELP, EXPN, VRFY и RSET **могут** использоваться любой момент на протяжении всего сеанса и даже без предварительной организации сеанса. Серверам SMTP **следует** нормально обрабатывать эти команды (т. е., не выдавать в ответ отклик 503) даже в тех случаях, когда эти команды используются до получения команды EHLO; клиентам **следует** открывать сессию с помощью команды EHLO до ввода перечисленных команд.

Если следовать этим правилам, пример из RFC 821, показывающий отклик «550 access denied to you» в ответ на команду EXPN некорректен, если команда EHLO не была введена до EXPN или клиенту не было отказано в обслуживании на основе IP-адреса клиента или по результатам аутентификации или аналогичных механизмов.

Команда MAIL (или устаревшие команды SEND, SOML, SAML) начинает почтовую транзакцию. После начала транзакции последняя включает начальную команду, одну или несколько команд RCPT и команду DATA, вводимые в указанном порядке. Почтовая транзакция прерывается командой RSET, новой командой EHLO или командой QUIT. В сеансе может происходить множество последовательных транзакций или не быть транзакций вообще. **Недопустимо** передавать команду MAIL (или SEND, SOML, SAML), если почтовая транзакция уже открыта, т. е., эту команду можно передавать только при отсутствии в сеансе продолжающейся почтовой транзакции – предыдущая транзакция должна быть завершена успешным выполнением команды DATA или прервана командой RSET или новой командой EHLO.

Если аргумент начинающей транзакцию команды неприемлем, **должен** возвращаться отклик 501 и сервер SMTP **должен** сохранять своё состояние. Если в сеансе нарушается порядок команд в такой степени, что это препятствует их выполнению сервером, последний должен возвратить отклик 503, сохраняя своё состояние.

Последней командой сеанса **должна** быть команда QUIT. Клиентам **следует** использовать команду QUIT для разрыва соединения даже в тех случаях, когда команда организации сеанса не была передана и воспринята.

4.1.5. Команды частного использования

Как было сказано в параграфе 2.2.2, команды, начинающиеся с X, могут использоваться в результате двухстороннего соглашения между клиентом (отправитель) и сервером (получатель) SMTP. Предполагается, что сервер SMTP, не распознающий такие команды, будет возвращать отклик 500 Command not recognized. Сервер SMTP с расширенными функциями **может** перечислить имена, связанные с командами частного использования, в своём отклике на команду EHLO.

Команды, переданные или воспринятые системами SMTP и не начинающиеся с X, **должны** соответствовать требованиям параграфа 2.2.2.

4.2. Отклики SMTP

Отклики на команды SMTP служат для синхронизации запросов и выполняемых действий при передаче почтовых сообщений, а также для обеспечения гарантии получения клиентом сведений о состоянии сервера SMTP. На каждую команду **должен** генерироваться единственный отклик.

Детальное описание последовательностей команда – отклик приводится в параграфе 4.3.

Отклик SMTP содержит трехзначный номер (передаётся как три числовых символа), за которым обычно следует строка текста, если в данной спецификации явно не указано иное. Числовые коды предназначены для автоматического определения состояния, в которое нужно перейти, текст – для человека. Цифровой код обеспечивает требуемую информацию и программе-клиенту не требуется просматривать текстовую часть отклика, которую в результате можно просто отбрасывать или передавать пользователю. Имеющиеся исключения из этого правила явно указаны в спецификации. В частности, коды откликов 220, 221, 251, 421 и 551 связаны с текстовыми сообщениями, которые клиентская программа должна разбирать и интерпретировать. В общем случае текст может зависеть от сервера или текущего контекста, т. е. каждый оклик может содержать разный текст. Обсуждение теоретических вопросов генерации откликов приводится в параграфе 4.2.1. Формально отклик определяется как последовательность: трехзначный код, <SP>, строка текста, <CRLF> или многострочный текст (см. параграф 4.2.1). Поскольку (в нарушение данной спецификации) текст иногда не включается в отклик, получившим такой отклик клиентам **следует** быть готовыми к обработке только числового кода (возможно, после кода в отклик будет помещён символ пробела). Предполагается, что лишь команды EHLO, EXPN и HELP могут возвращать многострочные отклики при нормальных обстоятельствах, однако такие отклики допускаются для всех команд.

В формате ABNF отклик сервера имеет вид:

```
Greeting       = ( "220 " (Domain / address-literal)
                 [ SP textstring ] CRLF ) /
                 ( "220-" (Domain / address-literal)
                 [ SP textstring ] CRLF
                 *( "220-" [ textstring ] CRLF )
                 "220" [ SP textstring ] CRLF )
textstring     = 1*(%d09 / %d32-126) ; HT, SP, Printable US-ASCII
Reply-line     = *( Reply-code "-" [ textstring ] CRLF )
                 Reply-code [ SP textstring ] CRLF
Reply-code     = %x32-35 %x30-35 %x30-39
```

где Greeting появляется только в откликах с кодом 220, анонсирующих открытие сервером своей части соединения. Другие возможные варианты откликов сервера на открытие соединения следуют синтаксису Reply-line.

Серверам SMTP **следует** передавать только отклики с кодами, указанными в этой спецификации, сопровождая их текстом, указанным в примерах, когда это приемлемо.

Клиенты SMTP **должны** определять свои действия только на основе кода отклика, а не его текста (за исключением change of address 251 и 551, а при необходимости и 220, 221, 421). В общем случае клиент **должны** воспринимать любой текст и отклики без текста (хотя серверам **не следует** передавать откликов, содержащих только код). Пробел после кода отклика рассматривается как часть текста. По возможности клиенту SMTP **следует** проверять первую цифру кода отклика (индикация важности).

Приведённый ниже список кодов **недопустимо** рассматривать как неизменный. Хотя добавление новых кодов является редким и значимым событием (предпочтительно добавление новой информации в текстовую часть отклика),

новые стандарты и проекты стандартов могут добавлять коды откликов. Следовательно, отправители SMTP **должны** быть готовы к обработке кодов, не указанных в данной спецификации. Такая обработка **должна** основываться на интерпретации только первой цифры кода.

При отсутствии согласованных с клиентом расширений для серверов SMTP **недопустимо** передавать отклики, в которых первая цифра кода отличается от 2, 3, 4 или 5. Клиентам при получении таких кодов **следует** трактовать их как признак неисправимой ошибки и прерывать почтовую транзакцию.

4.2.1. Важность кодов отклика и теоретические вопросы

Каждая из трёх цифр кода отклика имеет свой уровень значимости. Первая цифра определяет успех, неудачу или незавершённость команды. Для простых клиентов SMTP или при получении неизвестного кода можно определить дальнейшие действия (продолжение, повтор, отказ и т. п.), ограничившись первой цифрой кода. Клиенты SMTP, которые хотят получить более точную информацию о происходящем (ошибка почтовой системы, некорректный синтаксис и т. п.), могут использовать вторую цифру кода. Третья цифра и дополнительная информация в отклике служат для предоставления наиболее подробных сведений.

Первая цифра кода может принимать 4 значения:

2yz – позитивный отклик о завершении
Запрошенная операция успешно завершена и могут вводиться новые команды.

3yz – позитивный промежуточный отклик
Команда была воспринята, но запрошенные действия пока не выполнены и сервер ждёт дополнительной информации. Клиенту SMTP следует передать другую команду, содержащую требуемые данные. Отклики этой группы используются в командах с последовательным выполнением (например, DATA).

4yz – негативный отклик о временных проблемах
Команда не принята и запрошенная операция не выполнена. Однако условия, не позволяющие выполнить команду, носят временный характер и операция может быть запрошена вновь. Отправителю следует вернуться к началу последовательности команд (если таковая была). Понятие «временный» является недостаточно строгим и взаимодействующие стороны (клиент и сервер SMTP) должны одинаково интерпретировать его. Для каждого отклика этой группы время может различаться, но клиенту SMTP **следует** продолжать попытки. Различия между временными и постоянными проблемами (коды 5yz) достаточно условны и отклики 4yz обычно возвращаются в тех случаях, когда возможен позитивный результат при повторе без изменения формы команды и свойств отправителя или получателя (т. е., команда просто может быть повторена без изменений).

5yz - негативный отклик о постоянных проблемах
Команда не принята и запрошенная операция не выполнена. Клиенту SMTP **не следует** просто повторять команду, поскольку она заведомо не будет выполнена. Некоторые «постоянные» проблемы могут быть решены корректировкой команд, поэтому пользователь (человек) может запросить у клиента SMTP повтора операции после корректировки команд или их порядка (например, после проверки корректности ввода или изменения параметров учётной записи).

Нет ничего дурного в том, что протокол FTP [34] использует сходную архитектуру откликов и коды SMTP основаны на модели FTP. Однако в SMTP используется модель «команда - отклик», тогда как командный протокол FTP является асинхронным; кроме того, принятая в FTP группа кодов 1yz не используется в модели SMTP.

Вторая цифра отклика показывает категорию ошибки:

x0z Синтаксис: отклик связан с синтаксической ошибкой (команда синтаксически корректна, но отклик не может быть отнесён к другим категориям, нереализованная команда, излишняя команда и т. п.).

x1z Информация: отклик на запрос информации (например, справка или состояние).

x2z Соединение: отклики, относящиеся к каналу передачи.

x3z Не задан.

x4z Не задан.

x5z Почтовая система: отклики показывают состояние принимающей почтовой системы по отношению к запрошенной передаче или другим действиям почтовой системы.

Третья цифра позволяет получить дополнительную информацию для каждой категории, заданной второй цифрой. Приведённый ниже список откликов иллюстрирует это подход. Текстовая часть отклика является скорее рекомендуемой, чем обязательной и может изменяться в соответствии со связанной с откликом командой. С другой стороны, коды откликов должны в точности соответствовать приведённой в этом разделе спецификации. При разработке программ-серверов не следует изобретать новые коды для незначительно отличающихся ситуаций – нужно выбрать и адаптировать наиболее подходящий код из числа определённых в спецификации.

Например, команды типа NOOP, при успешном завершении которых клиент SMTP не получает новой информации, будут возвращать код 250. Отклик 502 возвращается при запросе нереализованной команды, а отклик 504 – для реализованных команд с неподдерживаемыми параметрами.

Текст отклика может содержать более одной строки и в таких случаях текст должен маркироваться так, чтобы клиент SMTP мог узнать о завершении текста. Это требует использования для многострочных откликов специального формата.

Формат многострочных откликов требует, чтобы каждая строка (кроме последней) начиналась кодом отклика, после которого следует дефис (-), а далее - текст. В последней строке вместо дефиса используется пробел - <SP>, после которого может следовать текст, и <CRLF>. Как указано выше, серверам следует передавать символ <SP>, если далее не будет текста, но клиент **должен** быть готов к отсутствию символа пробела.

Ниже приведён пример многострочного отклика:

```
250-Первая строка
250-Вторая строка
```

```
250-234 Текст, начинающийся с числа
250 Последняя строка
```

В многострочных откликах коды в каждой строке **должны** совпадать. Клиентам целесообразно принимать это во внимание и выполнять обработку с учётом кода из любой строки в предположении, что остальные строки содержат тот же код. В некоторых случаях важные для клиента данные передаются в тексте отклика. Клиент должен быть способен идентифицировать такие ситуации из текущего контекста.

4.2.2. Коды откликов (по группам)

500	Syntax error, command unrecognized	Синтаксическая ошибка, команда не распознана (это может говорить о слишком длинной команде).
501	Syntax error in parameters or arguments	Синтаксическая ошибка в параметрах или аргументах.
502	Command not implemented	Команда не реализована (см. параграф 4.2.4).
503	Bad sequence of commands	Некорректный порядок команд.
504	Command parameter not implemented	Параметр команды не реализован.
211	System status, or system help reply	Отклик с системной справкой или состоянием системы.
214	Help message	Информация о работе с сервером или отдельных командах.
220	<domain> Service ready	Служба для указанного домена готова.
221	<domain> Service closing transmission channel	Закрывается канал передачи для указанного домена.
421	<domain> Service not available, closing transmission channel	Для указанного домена обслуживание невозможно и канал связи закрывается. Это может быть откликом на любую команду, если известно, что сервис должен быть отключён.
250	Requested mail action okay, completed	Операция благополучно завершена.
251	User not local; will forward to <forward-path>	Нелокальный пользователь – почта будет пересылаться по прямому пути (см. параграф 3.4).
252	Cannot VRFY user, but will accept message and attempt delivery	Не удаётся проверить почтовый ящик, но сообщение принято и сервер попытается его доставить (см. параграф 3.5.3).
455	Server unable to accommodate parameters	Сервер не может принять параметры.
555	MAIL FROM/RCPT TO parameters not recognized or not implemented	Параметры команды MAIL FROM или RCPT TO не удалось распознать или их поддержка не реализована.
450	Requested mail action not taken: mailbox unavailable	Запрошенная операция невозможна – почтовый ящик недоступен (например, занят или временно заблокирован в соответствии с политикой).
550	Requested mail action not taken: mailbox unavailable	Запрошенная операция невозможна – почтовый ящик недоступен (например, почтовый ящик не найден, к нему нет доступа или команда отвергнута в соответствии с заданной политикой).
451	Requested action aborted: error in processing	Запрошенная операция прервана в результате ошибки.
551	User not local; please try <forward-path>	Нелокальный пользователь – попытайтесь использовать прямой путь (см. параграф 3.4).
452	Requested action not taken: insufficient system storage	Запрошенная операция не выполнена по причине нехватки пространства (на диске).
552	Requested mail action aborted: exceeded storage allocation	Запрошенная операция прервана по причине превышения выделенного (дискового) пространства.
553	Requested action not taken: mailbox name not allowed	Запрошенная операция не выполнена – недопустимый почтовый ящик (например, синтаксическая ошибка в имени ящика).
354	Start mail input; end with <CRLF>.<CRLF>	Начало ввода данных. Завершение по <CRLF>.<CRLF>
554	Transaction failed или No SMTP service here	Отказ транзакции или отсутствие поддержки сервиса SMTP (при попытке соединения)

4.2.3. Коды откликов в порядке номеров

211	System status, or system help reply	Отклик с системной справкой или состоянием системы.
214	Help message	Информация о работе с сервером или отдельных командах.
220	<domain> Service ready	Служба для указанного домена готова.
221	<domain> Service closing transmission channel	Закрывается канал передачи для указанного домена.
250	Requested mail action okay, completed	Операция благополучно завершена.
251	User not local; will forward to <forward-path>	Нелокальный пользователь – почта будет пересылаться по прямому пути (см. параграф 3.4).
252	Cannot VRFY user, but will accept message and attempt delivery	Не удаётся проверить почтовый ящик, но сообщение принято и сервер попытается его доставить (см. параграф 3.5.3).
354	Start mail input; end with <CRLF>.<CRLF>	Начало ввода данных. Завершение по <CRLF>.<CRLF>.
421	<domain> Service not available, closing transmission channel	Для указанного домена обслуживание невозможно и канал связи закрывается. Это может быть откликом на любую команду, если известно, что сервис должен быть отключён.
450	Requested mail action not taken: mailbox unavailable	Запрошенная операция невозможна – почтовый ящик недоступен (например, занят или временно заблокирован в соответствии с политикой).
451	Requested action aborted: error in processing	Запрошенная операция прервана в результате ошибки.
452	Requested action not taken: insufficient system storage	Запрошенная операция не выполнена по причине нехватки пространства (на диске).
455	Server unable to accommodate parameters	Сервер не может принять параметры.
500	Syntax error, command unrecognized	Синтаксическая ошибка, команда не распознана (это может говорить о слишком длинной команде).
501	Syntax error in parameters or arguments	Синтаксическая ошибка в параметрах или аргументах.

502	Command not implemented	Команда не реализована (см. параграф 4.2.4).
503	Bad sequence of commands	Некорректный порядок команд.
504	Command parameter not implemented	Параметр команды не реализован.
550	Requested mail action not taken: mailbox unavailable	Запрошенная операция невозможна – почтовый ящик недоступен (например, почтовый ящик не найден, к нему нет доступа или команда отвергнута в соответствии с заданной политикой).
551	User not local; please try <forward-path>	Нелокальный пользователь – попытайтесь использовать прямой путь (см. параграф 3.4).
552	Requested mail action aborted: exceeded storage allocation	Запрошенная операция прервана по причине превышения выделенного (дискового) пространства.
553	Requested action not taken: mailbox name not allowed	Запрошенная операция не выполнена – недопустимый почтовый ящик (например, синтаксическая ошибка в имени ящика).
554	Transaction failed или No SMTP service here	Отказ транзакции или отсутствие поддержки сервиса SMTP (при попытке соединения).
555	MAIL FROM/RCPT TO parameters not recognized or not implemented	Параметры команды MAIL FROM или RCPT TO не удалось распознать или их поддержка не реализована.

4.2.4. Отклик 502

У разработчиков часто возникают вопросы об использовании отклика 502 (Command not implemented – команда не реализована). Код 502 **следует** использовать в тех случаях, когда сервер SMTP распознал команду, но не умеет её выполнять. Если команда не распознана, **следует** возвращать код 500. Для систем SMTP с расширенными функциями в откликах на команду EHLO **недопустимо** указывать команды, приводящие к отклику 502 или 500.

4.2.5. Коды откликов после DATA и последующих <CRLF>.<CRLF>

Когда сервер SMTP возвращает позитивный отклик (код 2yz) после завершения команды DATA с последовательностью <CRLF>.<CRLF>, этот сервер принимает на себя ответственность за следующие операции:

- доставка сообщения (если почтовый ящик получателя существует);
- при неудачной попытке доставки в результате временных проблем предпринимается разумное число повторных попыток с перерывами (см. параграф 4.5.4);
- при неудаче вследствие долгосрочных проблем или после исчерпания заданного числа попыток в случае временных проблем исходному отправителю сообщения посылается уведомление (по адресу из команды MAIL).

Когда сервер SMTP возвращает отклик о временных проблемах (4yz) после команды DATA с завершающей последовательностью <CRLF>.<CRLF>, **недопустимо** предпринимать какие-то последующие попытки доставки этого сообщения. Клиент SMTP сохраняет за собой ответственность за доставку этого сообщения и может возвратить его пользователю или снова поставить в очередь на доставку (см. параграф 4.5.4.1).

Пользователю, отправившему сообщение, **следует** предоставить возможность интерпретировать характер проблем (временные или постоянные), передав ему сообщение по электронной почте или иным способом. Если клиент SMTP смог решить проблему доставки самостоятельно, уведомление пользователю не передаётся.

Когда сервер SMTP возвращает информацию о долгосрочных проблемах (код 5yz) после выполнения команды DATA с завершающей последовательностью <CRLF>.<CRLF>, **недопустимо** предпринимать какие-либо дополнительные попытки доставки сообщения. Как и для случая временных проблем, ответственность за доставку сообщения сохраняется за клиентом SMTP, но клиенту **не следует** пытаться повторить доставку тому же серверу без просмотра сообщения пользователем и внесения соответствующих изменений.

4.3. Порядок следования команд и откликов

4.3.1. Обзор

Связь между отправителем и получателем в процессе представляет собой диалог, контролируемый отправителем. Отправитель вводит команды, а получатель возвращает отклики на них. Если не согласованы другие условия с использованием расширенного сервиса, отправитель **должен** получить отклик на переданную команду прежде, чем посылать следующую. Одним из важных откликов является приветствие при организации соединения. Обычно получатель передаёт отклик 220 Service ready при завершении организации соединения. Отправителю **следует** дождаться этого отклика и только потом передавать следующие команды.

Примечание. Все отклики-приветствия включают официальное имя хоста (FQDN), на котором работает сервер, в качестве первого слова после кода. В некоторых случаях у хоста может не быть собственного имени. Обсуждение альтернативных имён для таких ситуаций приводится в параграфе 4.1.3.

Ниже приведены 3 примера приветствий:

```
220 ISIF.USC.EDU Service ready
220 mail.example.com SuperSMTP v 6.1.2 Service ready
220 [10.0.0.1] Clueless host service ready
```

В приведённой ниже таблице даны варианты откликов при удачном и неудачном завершении каждой команды. **Следует** строго придерживаться этих кодов. Получатель **может** изменять текст отклика, но смысл отклика и действия в ответ на него определяются числовым кодом и последовательностью введённых ранее команд и **должны** быть одинаковы.

4.3.2. Последовательности команда - отклик

Для каждой команды указаны обычные позитивные отклики. Используемые перед позитивными откликами префиксы включают I (промежуточный), S (успех) и E (ошибка). Поскольку некоторые серверы могут генерировать иные отклики в

соответствующих обстоятельствах и с учётом возможности появления новых кодов, клиентам SMTP **следует** (по возможности) интерпретировать только первую цифру кода. Кроме того, клиент **должен** быть готов к работе с неизвестными кодами, также интерпретируя в них только первую цифру. За исключением расширений, использующих механизмы, описанные в параграфе 2.2, для серверов SMTP **недопустима** передача кодов, содержащих что-либо сверх 3 цифр или использующих цифры, не входящие в разрешенный диапазон 2 - 5 (включительно).

Описанные здесь варианты откликов на команды (в принципе, и сами коды) могут дополняться или изменяться при использовании расширений SMTP, предлагаемых сервером и понятных (запрашиваемых) клиентом. Однако, если целью является более чёткая грануляция кодов, а не определение кодов для совершенно новых случаев, **следует** использовать систему, описанную в RFC 3463 [25], чтобы не изобретать новых кодов.

В дополнение к перечисленным в таблице кодам любые команды SMTP могут возвращать три приведённых ниже кода в соответствующих нештатных ситуациях:

500 - для случая command line too long (слишком длинная команда) или при получении непонятной команды. Отметим, что отклик command not recognized (неизвестная команда) в ответ на команду из обязательного набора является нарушением данной спецификации. Аналогично, генерация сообщения command too long в ответ на команду, длина которой не превышает 512 будет нарушением требований параграфа 4.5.3.1.4.

501 Syntax error in command or arguments (синтаксическая ошибка в команде или аргументах). Для поддержки будущих расширений командам, включённым в данную спецификацию, как команды без аргументов (DATA, RSET, QUIT), **следует** возвращать отклик 501 при получении команды с аргументами, если иное не согласовано в анонсированном EHLO расширении.

421 Service shutting down and closing transmission channel - сервис отключён. с разрывом коммуникационного канала.

В нормальных условиях в ответ на команды могут возвращаться следующие отклики:

Команда	Успех (S)	Неудача (E)
Организация соединения	220	554
EHLO или HELO	250	504[1], 550, 502[2]
MAIL	250	552, 451, 452, 550, 553, 503, 455, 555
RCPT	250, 251[3]	550, 551, 552, 553, 450, 451, 452, 503, 455, 555
DATA (промежуточный отклик 354)	250	552, 554, 451, 452, 450, 550 (отказ в соответствии с политикой)
DATA		503,55
RSET	250	
VRFY	250, 251, 252	550, 551, 553, 502, 504
EXPN	250, 252	550, 500, 502, 504
HELP	211, 214	502, 504
NOOP	250	
QUIT	221	

4.4. Трассировочная информация

Когда сервер SMTP получает сообщение для доставки или дальнейшей обработки, он **должен** вставить трассировочную информацию (time stamp или Received) в начало содержимого, как описано в параграфе 4.1.1.4.

Трассировочная строка **должна** иметь следующую структуру:

- В поле FROM, которое **должно** обеспечиваться в среде SMTP, **следует** включать (1) имя хоста-отправителя, представленное в команде EHLO, и (2) IP-адрес отправителя, определённый из соединения TCP.
- Поле ID **может** включать @, как предложено в RFC 822, но это необязательно.
- Поле FOR (если оно присутствует) **должно** содержать список элементов <path>, даже при использовании множества команд RCPT. Это может влиять на безопасность системы, поэтому нежелательно включать такие списки (см. параграф 7.2).

Для почтовых программ Internet **недопустимо** внесение изменений в строки Received:, уже присутствующие в заголовке сообщения. Серверы SMTP **должны** добавлять в начало свою строку Received, но **недопустимо** менять порядок имеющихся строк или вставлять свою строку Received в другое место.

По мере расширения сети Internet просмотр строк Received становится все более важным средством диагностики почтовых систем, особенно для обнаружения медленно работающих трансляторов. Серверам SMTP, которые создают поля Received, **следует** явно задавать временной сдвиг (например, -0800), а не использовать имена часовых поясов. По возможности следует указывать локальное время (с учётом пояса), а не UT[4]. Такая информация даёт больше сведений о локальных условиях. Если требуется использовать UT, получателю достаточно использовать простую арифметику для получения нужного значения. Использование формата UT приводит к потере информации о часовом поясе сервера. Если желательно указывать имя часового пояса, его **следует** давать как комментарий.

Когда сервер SMTP обеспечивает «окончательную доставку» сообщения, он вставляет строку обратного пути (return-path) в начало почтовых данных. Использование return-path является обязательным и почтовые системы **должны** поддерживать это. Строка обратного пути сохраняет значение параметра <reverse-path> из команды MAIL. Окончательная доставка сообщения все ещё сохраняет его в среде SMTP. Обычно сообщение доставляется в почтовый ящик пользователя или почтовое хранилище, но в некоторых случаях сообщение может подвергаться дополнительной обработке или передаваться другими почтовыми системами.

Адрес почтового ящика в пути возврата может отличаться от адреса отправителя. Например, сообщения об ошибках могут доставляться не отправителю, а направляться по специальному адресу для обработки. При использовании

[1]Соответствующая спецификации реализация будет возвращать такой код только в совершенно непонятных случаях.
[2]Разрешен только для старых серверов, которые не поддерживают EHLO.
[3]См. параграф 3.4, в котором обсуждается использование откликов 251 и 551
[4]Универсальное время. *Прим. перев.*

списков рассылки такой подход является общепринятым и весьма полезным, поскольку сообщения об ошибках направляются администратору списка, а не отправителю исходного письма.

В приведённом выше тексте предполагается, что окончательные почтовые данные будут начинаться со строки обратного пути, за которой будет следовать одна или несколько строк с временными метками. После этих строк будет следовать оставшаяся часть почтовых данных - почтовый заголовок и тело сообщения (RFC 5322 [4]).

В некоторых случаях серверу SMTP сложно определить обеспечивает ли он окончательную доставку, поскольку пересылка и другие операции могут происходить после восприятия сообщения для доставки. Поэтому все последующие почтовые системы (трансляторы, шлюзы, системы пересылки) **могут** удалять строку обратного пути и перестраивать команду MAIL, обеспечивая в доставленном сообщении единственную строку обратного пути.

Генерирующей сообщение системе SMTP **не следует** передавать сообщений, в заголовок которых уже включено поле Return-path. Для серверов SMTP, обеспечивающих трансляцию, **недопустимо** проверять данные сообщения и, особенно, наличие поля заголовка Return-path. Сервер SMTP, обеспечивающий окончательную доставку, **может** удалять имеющийся заголовок Return-path перед добавлением своего.

Основным назначением Return-path является указание адреса, по которому следует доставлять сообщения об ошибках. Для однозначности **следует** включать в сообщение единственный вариант обратного пути. Системам, использующим синтаксис RFC 822 с отличным от SMTP транспортом, **следует** указывать однозначный адрес, связанный с транспортным конвертом, по которому должна возвращаться информация об ошибках (например, о невозможности доставки).

Историческое замечание. Приведённые в RFC 822 сведения, отвергающие использование заголовка Return-path (или адреса возврата из команды MAIL в конверте) для доставки информации об ошибках, неприменимы в среде Internet. Адрес обратного пути (копируемый в Return-path) **должен** использоваться для доставки всех сообщений об ошибках в процессе доставки.

В частности:

- Шлюзам из SMTP в другие среды **следует** вставлять обратный путь, если у них нет информации о том, что другая среда также использует в адресах доменные имена Internet и поддерживает отдельный конверт с адресом отправителя.
- Шлюзам из других сред в SMTP **следует** удалять из заголовка строку обратного пути и копировать эту информацию в конверт SMTP или объединять её с присутствующей в конверте информацией из другой транспортной системы для построения обратного пути для команды MAIL в конверте SMTP.

Сервер должен принимать специальные меры в случаях, когда обработка принятых почтовых данных успешна лишь отчасти. Это может произойти, если после приёма нескольких адресов получателей и почтовых данных для них сервер SMTP обнаружит, что возможна доставка только некоторым из указанных адресатов. В таких случаях на команду DATA **должен** возвращаться отклик OK. Однако сервер SMTP **должен** подготовить и передать уведомление о невозможности доставки отправителю сообщения.

Должно передаваться одно уведомление со списком всех адресатов, которым невозможно передать сообщение, или отдельные уведомления для каждого из таких адресатов. Из соображений экономии **следует**, по возможности, использовать первый вариант. Отметим, что основная разница между обработкой псевдонимов (параграф 3.9.1) и пересылкой (данный параграф) состоит в изменении в данном случае обратного адреса. Все уведомления о невозможности доставки передаются с использованием команды MAIL (даже в тех случаях, когда проблема возникла при обработке устаревших команд SEND, SOML или SAML) и должны содержать пустое поле обратного пути (см. параграф 3.6).

Временные метки и пути возврата формально определяются следующим образом (определения FWS и CFWS даны в RFC 5322 [4]):

```
Return-path-line  = "Return-Path:" FWS Reverse-path <CRLF>
Time-stamp-line  = "Received:" FWS Stamp <CRLF>
Stamp          = From-domain By-domain Opt-info [CFWS] ";" FWS date-time
                 ; date-time определено в RFC 5322 [4], но формы «obs-», особенно для лет,
                 ; обозначенных 2 цифрами, запрещены в SMTP и их использование недопустимо.
From-domain    = "FROM" FWS Extended-Domain
By-domain      = CFWS "BY" FWS Extended-Domain
Extended-Domain  = Domain / ( Domain FWS "(" TCP-info ")" )
                   / ( address-literal FWS "(" TCP-info ")" )
TCP-info       = address-literal / ( Domain FWS address-literal )
                 ; сервер берет информацию из соединения TCP, а не из клиентской команды EHLO.
Opt-info       = [Via] [With] [ID] [For] [Additional-Registered-Clauses]
Via            = CFWS "VIA" FWS Link
With           = CFWS "WITH" FWS Protocol
ID             = CFWS "ID" FWS ( Atom / msg-id )
                 ; msg-id определено в RFC 5322 [4]
For            = CFWS "FOR" FWS ( Path / Mailbox )
Additional-Registered-Clauses  = CFWS Atom FWS String
                                 ; В этом месте могут добавляться определения из новых
                                 ; стандартов, зарегистрированные в IANA. Серверам SMTP
                                 ; не следует использовать незарегистрированные имена.
                           ; См. раздел 8.
Link           = "TCP" / Addtl-Link
Addtl-Link     = Atom
                 ; Дополнительные стандартные имена каналов, зарегистрированные IANA. Via -
                 ; предварительное значение для транспорта, отличного от Internet. Серверам
                 ; SMTP не следует использовать незарегистрированные имена.
Protocol       = "ESMTP" / "SMTP" / Attdl-Protocol
Attdl-Protocol = Atom
                 ; Дополнительные стандартные имена для протоколов, зарегистрированных IANA
```

```
; в реестре mail parameters [9]. Серверам SMTP не следует использовать
  ; незарегистрированные имена.
```

4.5. Другие вопросы реализации

4.5.1. Минимальная реализация

Для обеспечения работы SMTP **должна** быть обеспечена по крайней мере минимальная функциональность. Ниже перечислены команды, которые каждая реализация **должна** поддерживать в соответствии с данной спецификацией:

```
EHLO
HELO
MAIL
RCPT
DATA
RSET
NOOP
QUIT
VRFY
```

Любые системы, которые включают сервер SMTP, поддерживающий трансляцию или доставку почты, **должны** поддерживать зарезервированный почтовый ящик postmaster, как независимое от регистра символов локальное имя. Без такого адреса можно обойтись, если сервер всегда возвращает отклик 554 на открытие соединений (см. параграф 3.1). Требование принимать почту для адресата postmaster, ведёт к тому, что команды RCPT, указывающие адрес postmaster в любом из доменов, для которых сервер SMTP обеспечивает почтовое обслуживание, а также специальный случай команды RCPT TO:<Postmaster> (без указания домена), **должны** поддерживаться сервером.

Предполагается, что системы SMTP будут прилагать все разумные усилия для восприятия почты в адрес Postmaster от любой другой системы в Internet. В экстремальных случаях (например, при атаках на службы - DoS) или при нарушениях системы безопасности сервер SMTP может блокировать почту, направленную по адресу Postmaster. Однако, продолжительность такой блокировки **следует** максимально ограничивать, во избежание блокировки сообщений, которые не являются частью атаки.

4.5.2. Прозрачность

Без принятия некоторых специальных мер последовательность <CRLF>.<CRLF> будет восприниматься как завершение почтовых данных и не может включаться пользователем в текст. Обычно пользователи даже не знают о таких «запрещённых» последовательностях. Для прозрачной передачи подготовленного пользователем текста служат следующие процедуры:

- Перед отправкой строки почтового текста клиент SMTP проверяет первый символ строки. Если таким символом является точка, клиент просто добавляет к ещё одну точку в начале строки.
- Сервер SMTP, проверяет полученную строку. Если она содержит только точку, это трактуется как завершение данных. Если после точки в начале строки следуют дополнительные символы, эта точка просто удаляется.

Почтовые данные могут включать любые из 128 символов ASCII. Все символы доставляются в почтовый ящик получателя, включая пробелы, табуляторы и другие управляющие символы. Если канал передачи поддерживает поток данных в форме 8-битовых байтов (октетов), 7-битовые коды ASCII передаются с выравниванием по правому краю октета и нулевым значением старшего бита. Трансляторы SMTP используют специальную трактовку 8-битовых символов (см. 3.6).

В некоторых системах может требоваться передача данных в том виде, как они были приняты и сохранены. Это может быть актуально для хостов, использующих отличный от ASCII локальный набор символов, если они сохраняют данные в записях, а не в строках, или при использовании специальных символьных последовательностей в качестве ограничителей (delimiters) внутри почтовых ящиков. Если такие преобразования требуются, они **должны** быть обратимыми, особенно для почтовых трансляторов.

4.5.3. Размеры и тайм-ауты

4.5.3.1. Ограничения размеров

Существуют некоторые объекты, для которых требуется ограничение размера. Каждая реализация **должна** быть способна принимать объекты, размеры которых не выходят за эти ограничения. **Следует** (по возможности) избегать передачи объектов большего размера. Однако некоторые почтовые системы Internet создают такие адреса в формате X.400 (RFC 2156 [35]), которые могут потребовать большего размера объектов. Клиенты **могут** пытаться передать такие объекты, но они **должны** быть готовы к отказу серверов от обслуживания слишком больших объектов. Для снижения вероятности возникновения проблем в реализациях следует использовать методы, не ограничивающие размеры объектов.

Расширения SMTP могут включать поддержку символов, размер которых превышает 1 октет. По этой причине задаваемые в этом параграфе ограничения устанавливаются в октетах, а не в символах.

4.5.3.1.1. Локальная часть

Максимальный размер имени пользователя или локальной части адреса составляет 64 октета.

4.5.3.1.2. Домен

Максимальный размер доменного имени составляет 255 октетов.

4.5.3.1.3. Путь

Максимальная длина прямого и обратного пути составляет 256 октетов (включая разделители и пунктуацию).

4.5.3.1.4. Строка команды
Максимальная длина командной строки с учётом завершающей последовательности <CRLF> составляет 512 октетов. Расширения SMTP могут разрешать более длинные команды.

4.5.3.1.5. Строка отклика
Максимальная длина строки отклика с учётом кода и <CRLF> составляет 512 октетов. Дополнительную информацию можно передать, используя многострочный отклик.

4.5.3.1.6. Строка текста
Максимальная длина строки текста с учётом <CRLF> составляет 1000 октетов (без учёта добавляемую для обеспечения прозрачности точки в начале). Расширения SMTP могут использовать более длинные строки.

4.5.3.1.7. Содержимое письма
Ограничение максимального размера содержимого сообщения (включая заголовки и тело) **должно** быть не менее 64K октетов. После введения стандартов Internet на multimedia-почту (RFC 2045 [21]) размеры почтовых сообщений Internet многократно возросли и по возможности следует избегать ограничения размера сообщений. Системам SMTP, которые не могут отказаться от ограничения размеров **следует** реализовать сервисное расширение SIZE (RFC 1870 [10]), а клиентам SMTP, передающим большие сообщения, **следует** по возможности использовать это расширение.

4.5.3.1.8. Буфер адресатов
Минимальное число буферизуемых получателей **должно** составлять 100. Отказ от приёма сообщений (для избыточных получателей) при числе команд RCPT менее 100 является нарушением данной спецификации. Для транслирующих серверов SMTP **недопустимо**, а доставляющим серверам **не следует** проверять число адресатов в заголовке и отвергать сообщения на основе общего числа получателей. Сервер, в котором ограничивается число получателей, **должен** использовать разумный выбор отклоняемых сообщений (скорее сразу отклонить адресатов, выходящих за пределы допустимого числа, нежели потом отбрасывать принятые сначала адреса). Клиентам, которым требуется доставка сообщения, включающего более 100 команд RCPT, **следует** быть готовыми к передаче блоками по 100 адресов в один приём.

4.5.3.1.9. Трактовка выхода за пределы
Ошибки, связанные с выходом за допустимые пределы, приводят к передаче соответствующих откликов:

```
500 Line too long – слишком длинная строка
501 Path too long – слишком длинный путь
452 Too many recipients – слишком много получателей (см. ниже)
552 Too much mail data – слишком много почтовых данных.
```

4.5.3.1.10. Слишком много получателей
В RFC 821 [1] некорректно указано, что сервер SMTP в случаях превышения числа команд RCPT (too many recipients) генерирует отклик с кодом 552. Корректным кодом для таких откликов является 452. Клиентам **следует** трактовать код 552 в таких случаях как временную проблему, а не постоянную, чтобы описанная ниже логика могла работать.

Когда соответствующий спецификации SMTP сервер сталкивается с такой проблемой, он имеет по крайней мере 100 принятых команд RCPT в своём буфере получателей. Если сервер способен принять сообщение, из клиентской очереди будет удалено по крайней мере 100 адресов. Когда клиент предпримет новую попытку передачи адресов, для которых был получен отклик 452, сервер SMTP сможет поместить в буфер получателей по крайней мере 100 адресов. Каждая повторная попытка будет обеспечивать передачу сообщения по крайней мере сотне адресатов.

Если сервер SMTP имеет предел для числа команд RCPT и этот предел превышен, сервер **должен** использовать отклик с кодом 452 (но клиенту **следует** быть готовым и к получению кода 552, как было указано выше). Если ограничения сервера заданы правилами, он **может** использовать отклик с кодом 5yz. В частности, если задача состоит в том, чтобы запретить передачу сообщений с числом получателей, превышающим заданное для сайта значение, а не просто ограничить число адресатов для данной почтовой транзакции, разумно будет возвращать отклик 503 на любую команду DATA, полученную после отклика 452 (или 552), или просто возвращать код 503 после команды DATA без предшествующего негативного отклика.

4.5.3.2. Тайм-ауты
Клиенты SMTP **должны** поддерживать механизм тайм-аутов. Тайм-ауты **должны** задаваться для команд, а не для времени всей почтовой транзакции. **Следует** обеспечивать возможность настройки значений параметров без повторной компиляции кода SMTP. Для реализации этих требований таймеры задаются независимо для каждой команды SMTP и каждого буфера передачи данных. Последнее означает, что общий тайм-аут для транзакции растёт пропорционально увеличению размера сообщения.

На основе опыта работы трансляторов с высокой нагрузкой значения тайм-аутов определены приведённые ниже значения, которые **следует** использовать.

4.5.3.2.1. Стартовое сообщение 220: 5 минут
Клиентский процесс SMTP должен отличать сбои в соединениях TCP от задержки получения стартового приветствия с кодом 220. Многие серверы SMTP воспринимают соединение TCP, но задерживают передачу отклика 220, пока в системе не освободится достаточное для обработки почты количество ресурсов.

4.5.3.2.2. Команда MAIL: 5 минут

4.5.3.2.3. Команда RCPT: 5 минут
Если обработка списков рассылки и псевдонимов не откладывается до приёма сообщения, требуется увеличение тайм-аута.

4.5.3.2.4. Инициирование команды DATA: 2 минуты
Этот тайм-аут определяет время ожидания отклика 354 Start Input на команду DATA.

4.5.3.2.5. Блок данных: 3 минуты
Время ожидания завершения каждого вызова TCP SEND для передачи блока данных.

4.5.3.2.6. Прерывание команды DATA: 10 минут
Время ожидания отклика 250 OK. Когда получатель принимает завершающую сообщение точку, он обычно начинает обработку полученных данных для доставки сообщения в почтовый ящик пользователя. Ложные тайм-ауты в это время крайне нежелательны и обычно приводят к доставке многочисленных копий, поскольку сообщение может быть уже послано и сервер принял на себя ответственность за его доставку (см. параграф 6.1).

4.5.3.2.7. Тайм-аут сервера: 5 минут
Серверам SMTP **следует** использовать тайм-аут не менее 5 минут при ожидании от клиента следующей команды.

4.5.4. Стратегии повтора
Общая структура реализации хоста SMTP включает пользовательские почтовые ящики, одну или несколько областей для хранения очереди сообщений и один или несколько демонов, обслуживающих приём и передачу почты. Точная структура сильно зависит от потребностей пользователей хоста, а также числа и размера поддерживаемых хостом списков рассылки. Ниже описано несколько вариантов оптимизации, которые могут быть особенно полезны для почтовых хостов с большой нагрузкой.

Любая стратегия организации очередей **должна** включать тайм-ауты для всех операций, задаваемые независимо для каждой команды. При любых обстоятельствах **недопустимо** возвращать сообщения об ошибках в ответ на сообщения об ошибках.

4.5.4.1. Стратегия передачи
В рамках общей модели клиент SMTP представляет собой один или несколько процессов, которые периодически пытаются передавать исходящую почту. В типичной системе программы подготовки почтовых сообщений имеют тот или иной способ запроса немедленной обработки исходящей почты; почта, которая не может быть отправлена незамедлительно, **должна** помещаться в очередь, и отправитель будет периодически пытаться её отослать адресатам. Запись почтовой очереди будет включать не только само сообщение, но и конверт для его доставки.

Отправитель **должен** делать паузу перед повторной попыткой отправить почту адресату после неудачи. В общем случае интервал ожидания **следует** делать не менее 30 минут, однако более изощрённые подходы с переменным временем ожидания будут давать преимущества в тех случаях, когда клиент SMTP может определить причину неудачи.

Попытки продолжаются до тех пор, пока сообщение не будет доставлено или пока не истечёт заданное на попытки повтора время (обычно, не менее 4 - 5 дней). **Может** оказаться целесообразным выбор меньшего числа попыток передачи уведомлений о невозможности доставки и аналогичных сообщений об ошибках при сохранении обычного числа попыток для остальной почты. Параметры повтора **должны** быть настраиваемыми.

Клиенту **следует** сохранять список хостов, которым не удалось отправить почту, и соответствующие значения тайм-аутов для соединений вместо использования «тупых» попыток повтора.

Опыт показывает, что ошибки обычно носят временный характер (отсутствие связи с системой адресата или нарушение работы этой системы) и рекомендуется делать две попытки повтора в течение первого часа хранения письма в очереди, а далее повторять попытки передачи каждые 2 – 3 часа.

Клиент SMTP может сократить задержку перед повтором, согласовав такое совращение с сервером SMTP. Например, при получении почты с какого-то адреса очевидна возможность передачи по этому адресу почты из очереди (если она есть). Применяя такой подход, приложение в большинстве случаев может обойтись без явного использования функций «передать сообщения из очереди сейчас» типа ETRN (RFC 1985 [36]).

Возможна дальнейшая оптимизация стратегии передачи путём использования множества адресов на хосте (см. ниже), ускоряющего доставку почты за счёт повышения расхода ресурсов сервера.

Клиент SMTP может иметь большую очередь сообщений для каждого из недоступных хостов. Если все такие сообщения включать в каждую попытку повтора, это будет порождать значительный избыточный трафик в Internet, а почтовая система будет недоступна в течение длительного периода. Отметим, что клиент SMTP в общем случае может констатировать неудачную попытку только по истечении тайм-аута (несколько минут) и даже минутная задержка на соединение будет приводить к очень большим задержкам, если в очереди скопились десятки или даже сотни недоставленных сообщений для одного хоста.

В то же время, клиентам SMTP **следует** с большой осторожностью использовать кэшированные негативные отклики от серверов. В экстремальном случае, если команда EHLO вводится много раз в течение одного соединения SMTP, сервер может возвращать разные отклики. Очень важно подчеркнуть, что отклики 5yz на команду MAIL **недопустимо** кэшировать.

Когда сообщение доставляется множеству адресатов и сервер SMTP, на который копируется сообщение для передачи, совпадает для множества получателей, **следует** передавать единственную копию сообщения. Т. е., клиенту SMTP **следует** использовать последовательность команд MAIL, RCPT, RCPT,... RCPT, DATA вместо последовательности MAIL, RCPT, DATA, ..., MAIL, RCPT, DATA. Однако при большом количестве адресатов может быть превышено допустимое число повторов команды RCPT на одну команду MAIL. Описанный метод повышения эффективности **следует** реализовать.

Клиент SMTP для обеспечения своевременной доставки **может** поддерживать множество одновременных исходящих почтовых транзакций. Однако для предотвращения избыточного расхода ресурсов хоста на обработку почты, число одновременных транзакций может ограничиваться.

4.5.4.2. Стратегия приёма

Серверу SMTP **следует** пытаться сохранить постоянное прослушивание порта SMTP (в соответствии с реестром IANA порт 25). Это требуется для поддержки множества входящих TCP-соединений для SMTP. Некоторые ограничения **возможны**, но серверы, неспособные одновременно обслуживать множество транзакций SMTP, являются нарушением данной спецификации.

Как было сказано выше, сервер SMTP при получении почты от какого-либо из хостов может активизировать свой механизм очередей SMTP для попытки повторной передачи почты, хранимой для этого хоста.

4.5.5. Сообщения с пустым полем обратного пути

Некоторые типы уведомлений, требуемые существующими или предложенными стандартами, передаются с пустым полем обратного пути. К числу таких сообщений относятся уведомления об ошибках при доставке (см. параграф 3.7), другие типы сообщений DSN (Delivery Status Notifications – уведомления о состоянии доставки RFC 3461 [32]) и сообщения MDN (Message Disposition Notifications – уведомления о диспозиции сообщений RFC 3798 [37]). Все типы указанных сообщений являются уведомлениями о предыдущем сообщении и посылаются по обратному пути из заголовка письма, с которым связано данное уведомление. Невозможность доставки зачастую связана с проблемами в почтовой системе на хосте адресата, поэтому некоторые АДП настраиваются на пересылку таких уведомлений кому-нибудь, кто будет способен исправить проблему с почтой (например, с использованием псевдонима postmaster).

Все остальные типы сообщений (т. е., любые сообщения, для которых проекты стандартов RFC не требуют использовать пустой путь возврата) **следует** посылать с корректным, непустым полем обратного пути.

Разработчикам автоматизированных почтовых систем **следует** быть аккуратными и обеспечивать корректную обработку разных типов сообщений с пустым путём возврата. В частности, таким системам **не следует** отвечать на сообщения без обратного пути и **не следует** добавлять непустой путь возврата или менять пустой обратный путь на непустой при пересылке таких сообщений.

5. Преобразование адресов и обработка почты

5.1. Обнаружение целевого хоста

После того, как клиент SMTP лексически идентифицирует домен, для которого предназначена передаваемая на обработку почта (см. параграфы 2.3.5 и 3.6), **должно** выполняться обращение к серверу доменных имён (DNS lookup) для преобразования доменного имени (RFC 1035 [2]). Предполагается, что в адресах используются полные имена (FQDN) – механизм определения FQDN по частичному имени или локальному псевдониму выходит за пределы данной спецификации. В силу сложившейся практики, серверам SMTP, используемым для начального представления сообщений, **не следует** выполнять преобразований неполных имён (серверы представления сообщений [18] имеют более мощные механизмы таких преобразований), а для транслирующих серверов SMTP такие преобразования **недопустимы**.

При поиске сначала предпринимается попытка найти локальную запись MX, связанную с именем. Если взамен этого будет найдена запись CNAME, полученное в результате имя обрабатывается как исходное. Если сервер имён сообщает об отсутствии искомого домена, **должно** генерироваться сообщение об ошибке. При возврате сообщения о временной ошибке письмо **должно** помещаться в очередь для последующего повтора попытки передачи (см. параграф 4.5.4.1). Если возвращается пустой список записей MX, адрес трактуется, как связанный с неявной записью MX RR, имеющей уровень предпочтения 0 и указывающей на данный хост. Если записи MX присутствуют, но ни одна из них не может быть использована, или неявная запись MX не может быть использована, **должно** генерироваться сообщение об ошибке.

Если для данного имени найдена одна или несколько записей MX, для системы SMTP **недопустимо** использовать какие-либо адресные записи, связанные с этим именем, пока они не найдены с использованием записей MX; приведённое выше правило неявных записей MX применимо только в случаях отсутствия реальных MX. Если записи MX присутствуют, но ни одна из них не может быть использована, **должно** генерироваться сообщение об ошибке.

Когда найдено доменное имя, связанное с MX RR, и полученное соответствующее поле данных, этот отклик **должен** включать доменное имя. При запросе по этому имени **должна** возвращаться по крайней мере одна адресная запись (например, A или AAAA), которая даёт IP-адрес сервера SMTP для отправки тому сообщения.

Все прочие отклики, включая значения, возвращающие при запросе запись CNAME, выходят за пределы рассмотрения данного стандарта. Запрет меток, которые преобразуются в записи CNAME более подробно рассматривается в параграфе 10.3 RFC 2181 [38].

После успешного поиска доменного имени в DNS преобразование может дать не один адрес, а несколько, в результате наличия множества записей MX и/или поддержки хостом нескольких адресов (multihoming). Для обеспечения надёжной доставки почты клиент SMTP **должен** быть способен пытаться (включая повторы) использовать все адреса в соответствии с их порядком в списке, пока доставка не завершится успехом. Однако **может** существовать конфигурационное ограничение числа используемых альтернативных адресов. В таких случаях клиенту SMTP **следует** предпринимать попытки по крайней мере для двух адресов.

Для ранжирования адресов хостов используется два типа данных – множественные записи MX и многодомные хосты.

Записи MX содержат информацию о предпочтениях, которая **должна** использоваться при сортировке списка, если число записей превышает 1 (см. ниже). Меньшие значения MX указывают на более предпочтительные адреса доставки. При наличии нескольких адресов с одинаковыми значениями MX нет явных причин для предпочтения того или иного адреса и отправитель SMTP **должен** выбирать порядок таких адресов случайным образом для распределения нагрузки между разными почтовыми серверами одной организации.

Хост получателя (возможно с предпочтительной записью MX) может оказаться многодомным – в таких случаях доменное имя будет преобразовываться в список адресов IP. Ответственность за упорядочивание этого списка лежит на интерфейсе преобразователя имён (domain name resolver), который должен упорядочивать список в порядке снижения предпочтений, а отправитель SMTP **должен** пытаться использовать адреса в предложенном порядке.

Хотя поддержка попыток доставки с использованием множества адресов требуется от реализации, возможность таких попыток для конкретной инсталляции может быть ограничена или отключена совсем. Вопрос о целесообразности использования разных адресов многодомных хостов остаётся спорным. Основным аргументом в пользу таких попыток является повышение вероятности своевременной доставки сообщений, а в некоторых случаях – просто обеспечение возможности доставки. Противники такого подхода считают, что он ведёт к излишнему расходу ресурсов. Отметим, что использование ресурсов сильно зависит от выбранной стратегии передачи, как было показано в параграфе 4.5.4.1.

Если сервер SMTP принимает сообщение, для адресата которого данный сервер означен в записи MX, этот сервер **может** транслировать сообщение (потенциально, после получения переписанных адресов для MAIL FROM и/или RCPT TO), обеспечивая его окончательную доставку, или передать его дальше, используя тот или иной механизм, не относящийся к транспортной среде SMTP. Естественно, для второго случая сначала должен быть проверен список записей MX.

Если сервер определяет, что ему следует транслировать сообщение без переписывания заголовков, он **должен** отсортировать записи MX для определения нужной. Высший приоритет для передачи сообщения будет иметь запись с минимальным значением MX. Хост-транслятор **должен** проверить список на предмет наличия в нем имён или адресов, известных для данной транзакции. Если найдена соответствующая запись, все остальные записи, для которых уровень предпочтения не выше найденного, должны исключаться из рассмотрения. Если таких записей нет, это говорит об ошибке и сервер **должен** вернуть сообщение, как недоставляемое. С оставшимися в списке записями **следует** повторять попытки доставки сообщения в порядке снижения приоритета записи.

5.2. IPv6 и записи MX

В современной сети Internet клиенты и серверы SMTP могут использоваться на хостах IPv4, IPv6 или смешанных, которые поддерживают обе версии протокола IP. Домены хостов, на которые указывают записи MX, могут, следовательно, включать записи A RR (IPv4), AAAA RR (IPv6) и комбинации таких записей. Хотя в RFC 3974 [39] рассмотрен некоторый опыт использования в смешанных средах, этого опыта недостаточно для стандартизации, а некоторые из содержащихся в документе рекомендаций не вполне совместимы с настоящей спецификацией. Предпринимаемые действия могут зависеть от локальных условий (таких, как производительность участвующих в процессах сетей) и требуемых преобразований или быть обычными (например, клиентам, поддерживающим только IPv6, не нужно пытаться отыскать записи A RR или обращаться к серверам, которые поддерживают только IPv4). Разработчикам реализаций SMTP, которые могут работать в IPv6 или смешанных средах, следует изучить упомянутые выше процедуры, особенно комментарии по поводу многодомных хостов, и, предпочтительно, обеспечить механизмы, облегчающие тонкую настройку рабочей среды и взаимодействие почтовых систем IPv4 и IPv6 с учётом локальных условий.

6. Обнаружение и решение проблем

6.1. Гарантированная доставка и отклики по электронной почте

Когда получатель SMTP принимает порцию почты (передав отклик 250 OK в ответ на команду DATA), он принимает на себя ответственность за доставку или трансляцию сообщения. К этой ответственности следует относиться серьёзно. **Недопустима** потеря сообщений по незначительным причинам, типа последующего «падения» хоста или предсказуемой нехватки ресурсов. Некоторые серьёзные причины потери сообщений рассмотрены в следующем параграфе и параграфе 7.8.

Если после восприятия сообщения обнаруживается невозможность его доставки, получатель SMTP **должен** подготовить и передать уведомление об этом. При передаче уведомления **должен** использоваться пустой (<>) путь возврата в конверте. Получателем такого уведомления **должен** быть адрес из обратного пути в конверте (или строке Return-Path:). Если обратный путь пустой ("<>"), для сервера SMTP **недопустима** передача уведомления. Обычно, ничто не запрещает на локальном уровне (в той же среде, к которой относится получатель SMTP) принимать решение о протоколировании или иной фиксации сведений о пустом пути возврата. Если адресом является явный маршрут source route, из него **должен** выделяться последний интервал (final hop).

В качестве примера предположим, что нужно передать уведомление для сообщения, принятого по команде:

```
MAIL FROM:<@a,@b:user@d>
```

Уведомление **должно** передаваться с помощью команды:

```
RCPT TO:<user@d>
```

Некоторые проблемы с доставкой после того, как система SMTP восприняла сообщение, неизбежны. Например, у принимающего сервера SMTP может не быть возможности проверки всех адресов доставки в командах RCPT по причине некритических (soft) ошибок в системе доменных имён, поскольку адресатом является список рассылки (см. описание RCPT), или сервер действует как транслятор и не имеет непосредственного доступа к системе доставки.

Во избежание дублирования сообщений в результате тайм-аутов, получатель SMTP **должен** пытаться минимизировать время отклика на индикатор завершения данных <CRLF>.<CRLF>. Подробное обсуждение этого вопроса приводится в RFC 1047 [40].

6.2. Нежелательная и незапрошенная почта, почтовые атаки

Практичность и предсказуемость почтовой системы Internet требует, чтобы сообщения, которые могут быть доставлены, доставлялись на практике независимо от синтаксических ошибок и других отказов, связанных с сообщениями, а также независимо от содержимого почты. Если почта не может быть доставлена и не может быть отвергнута сервером SMTP в транзакции SMTP, эта почта должна быть возвращена (bounced) с уведомлением о невозможности доставки, как описано выше. В современном мире, когда многие операторы серверов SMTP убедились в том, что количество нежелательной почты большого размера во много раз превышает объем желаемой почты и когда восприятие сообщения может вызвать дополнительный нежелательный трафик, связанный с верификацией адреса, приведённый выше принцип утрачивает практический смысл.

Как сказано в параграфах 7.8 и 7.9, отбрасывание почты без уведомления отправителя на практике стало разрешённым. Однако такое поведение весьма опасно и нарушает сложившиеся за многие годы традиции, а также не

соответствует предположениям пользователей о том, что почта доставляется или приходит уведомление о невозможности доставки. Неразумное отбрасывание электронной почты без уведомления отправителей может свести на нет эффективность почтовой системы Internet. Поэтому отбрасывание почтовых сообщений без уведомления следует использовать только в тех случаях, когда есть веские основания считать сообщения мошенническими или неприемлемыми по иным причинам.

Рациональным путём усиления принципа обеспечения доставки, если таковая возможна, является отказ от доставки сообщений, имеющих некорректный адрес возврата. Однако опыт показывает, что пользователей больше устраивает вариант, когда доставляются все сообщения, доставка которых возможна. Надёжное определение некорректности адреса возврата может оказаться затруднительным и достаточно долгим процессом, особенно в тех случаях, когда предполагаемая почтовая система не доступна напрямую или не поддерживает полностью функциональность VRFY. Даже при выборе политики отбрасывания почты с некорректными обратными адресами, такую политику **следует** применять лишь в тех случаях, когда практически не остаётся сомнений в некорректности обратного адреса.

Если сообщение отвергается по причине неприемлемости его содержимого (критерии принятия такого решения выходят за пределы функциональности сервера SMTP, определённой в этом документе), уведомления об отказе от (bounce) **не следует** передавать, пока принимающая система не уверена в полезности таких уведомлений. Предпочтительным для таких ситуаций поведением (его следует принимать по умолчанию) является отказ от передачи уведомлений об отказе от доставки сообщений с неприемлемым содержанием.

6.3. Детектирование петель

Простой подсчёт числа заголовков Received: в принятых сообщениях обеспечивает эффективный, но обычно неоптимальный способ обнаружения петель в почтовой системе. Серверам SMTP, использующим такой способ, **следует** устанавливать высокий порог отказа (обычно, не менее 100 записей Received). Независимо от используемого механизма сервер **должен** обеспечивать средства детектирования и предотвращения тривиальных петель.

6.4. Компенсация отклонений от стандартов

К несчастью приходится сталкиваться со множеством вариаций, творческих реализаций и откровенных нарушений стандартов для почтовых протоколов Internet – одни возникают часто, другие - реже. Дебаты по вопросам политики корректно реализованных получателей (серверов) или трансляторов SMTP относительно некорректно подготовленных сообщений (пытаться передать их в неизменном виде, отвергнуть или пытаться исправить для повышения вероятности успешной доставки и последующего ответа на них) начались почти одновременно с появлением структурированной почты и конца этому обсуждению не видно. Сторонники жёсткой политики утверждают, что попытки исправления редко дают положительные результаты и отказ от передачи плохих сообщений является единственным способом избавиться от некорректно работающих почтовых программ. Сторонники исправления сообщений или доставки в неизменном виде считают, что пользователи предпочитают почту, которая работает во всех возможных ситуациях, и на этом направлении может существовать значительное давление рынка. На практике давление рынка может оказаться более сильным для отдельных производителей, нежели требования стандартов, независимо от наличия у фирсы реальных разработчиков.

Проблемы, связанные с некорректным форматом сообщений, обострились после введения специальных протоколов для чтения (загрузки) почты с серверов [POP2 [15], POP3 [16], IMAP2 [41], PCMAIL [42]). Эти протоколы поддерживают использование SMTP в качестве протокола передачи (представления сообщений) и серверов SMTP для трансляции почты на хосты клиентов этих протоколов (которые часто не имеют прямого постоянного подключения к Internet). Исторически многие из таких хостов не поддерживают часть механизмов и данных, используемых SMTP (и протоколом почтовых форматов RFC 822 [28]). Некоторые могут не сохранять значение текущего времени, другие не понимают часовых поясов, третьи не знают своего имени и, конечно, ни один из таких хостов не может удовлетворять тем требованиям, которые заложены в концепцию заверенных адресов RFC 822.

В ответ на появление «ущербных» клиентов SMTP многие системы SMTP сейчас дополнительно обрабатывают сообщения, полученные от таких клиентов в неполном или некорректном формате. Такая стратегия в общем случае приемлема, когда сервер может идентифицировать или аутентифицировать клиента и между клиентом и сервером существует предварительное соглашение. Такое решение значительно лучше по сравнению с исправлениями, которые могут вносить серверы доставки или трансляции для малознакомых или совсем неизвестных пользователей и клиентских машин. Многие из этих проблем решаются за счёт использования отдельного протокола для представления сообщений (типа RFC 4409 [18]) взамен использования для этих целей исходных серверов SMTP.

Ниже перечислены изменения, которые **могут** быть при необходимости внесены в обрабатываемые сообщения отправляющими (исходными) серверами SMTP или при использовании серверов SMTP для представления почты:

- добавление поля message-id при его отсутствии;
- добавление даты, времени и часового пояса при их отсутствии;
- корректировка адреса в соответствии с форматом FQDN.

Чем меньше информации сервер имеет от клиента, тем менее очевидны корректировки и больше осмотрительности и консерватизма следует использовать при рассмотрении вопроса о возможности и способах корректировки сообщений. Перечисленные выше изменения **недопустимо** выполнять на промежуточных трансляторах SMTP.

В любом случае корректно реализованные клиенты, предоставляющие корректную информацию будут иметь предпочтение при корректировке сообщений серверами SMTP. Во всех ситуациях рекомендуется тщательно документировать (в полях трассировки и/или комментариях в заголовке) вносимые сервером изменения.

7. Вопросы безопасности

7.1. Безопасность почты и обманки

Природа почты SMTP не обеспечивает безопасности, поскольку любой пользователь может напрямую взаимодействовать с принимающим или транслирующим сервером SMTP, создавать сообщения и обманывать

простодушных получателей, полагающих, что это почта от кого-то другого. Создание таких сообщений, обманный характер которых не обнаруживается, – задача более сложная, но вполне посильная для людей с нужными знаниями и соответствующей мотивацией. Следовательно, по мере повышения уровня знаний в сфере почты Internet человек начинает понимать, что почта SMTP не может быть заверена на транспортном уровне, равно как не обеспечивается и проверка целостности почты. Реальная безопасность почты обеспечивается только сквозными методами, включающими контроля тела сообщения, использования цифровых подписей (см. RFC 1847 [43] и, например, PGP[1] в RFC 4880 [44] или S/MIME[2] в RFC 3851 [45]).

Различные сервисные расширения и конфигурационные опции, которые обеспечивают аутентификацию на транспортном уровне (например, от клиента к серверу SMTP), несколько улучшают ситуацию. Однако, в общем случае, это ограничивается аутентификацией одного сервера другим, вместо аутентификации всей цепочки трансляторов и пользователей или пользовательских машин на серверах. Следовательно, пока эти методы не дополнены аккуратной передачей ответственности в хорошо спроектированной среде с доверительными отношениями, унаследованная уязвимость транспортной среды SMTP (по сравнению с механизмами сквозного использования цифровых подписей) будет сохраняться.

Попытки затруднить пользователям подстановку в поле обратного пути в конверте и заголовке From корректного чужого адрес взамен адреса реального отправителя заведомо ошибочны - это затрудняет работу легитимных приложений, в которых почта передаётся одним пользователем по просьбе другого или отклики на ошибки (доставку) должны передаваться по специальному адресу (системы, которые обеспечивают пользователю удобные способы замены этих полей для каждого сообщения отдельно, должны будут пытаться создать основные и постоянные адреса почтовых ящиков для пользователей в соответствии с полями Sender в генерируемых сообщениях).

Эта спецификация не рассматривает вопросы аутентификации, связанные в протоколом SMTP, но полезная функциональность не может ущемляться в надежде на несущественную защиту против фальсифицированной почты.

7.2. Скрытые копии - BC

В передаваемых серверу SMTP командах RCPT могут присутствовать адреса, по тем или иным причинам не указанные в заголовке сообщения. Двумя основными случаями являются использование почтового адреса, как «детонатора» списка (один адрес преобразуется во множество адресов реальных получателей) и скрытых копий (blind copy – или bc). В тех случаях, когда используется более одной команды RCPT и во избежание подавления некоторых функций этих механизмов, клиентам и серверам SMTP **не следует** копировать весь набор аргументов команды RCPT в заголовки, как часть трассировочных полей, информационных полей или заголовков частного расширения. Поскольку на практике это правило часто нарушается и его выполнение не может быть обеспечено принудительно, передающие системы SMTP, которые знают об использовании bcc, **могут** счесть полезной передачу каждой скрытой копии в отдельной транзакции с единственной командой RCPT.

Не существует неразрывных отношений между обратным (из команд MAIL, SAML и т. п.) или прямым (RCPT) адресом в транзакции SMTP (конверте) и адресами в заголовке. Принимающим системам **не следует** пытаться найти такие соотношения и использовать их для изменения заголовков с целью доставки сообщения. Популярный заголовок Apparently-to (видимо для) является нарушением данного принципа и хорошо известным случаем непреднамеренного разглашения информации; **не следует** пользоваться этим заголовком.

7.3. VRFY, EXPN, Security

Как обсуждалось в параграфе 3.5, отдельные сайты могут заблокировать использование команд VRFY и EXPN из соображений безопасности (см. ниже). Как следует из сказанного выше, для реализаций, обеспечивающих возможность такой блокировки, **недопустимо** показывать, что они могут проверять адреса, не проверяя их фактически. Если сайт блокирует команды из соображений безопасности, сервер SMTP **должен** возвращать отклик 252, а не код, который может ввести в заблуждение относительно результатов верификации адреса.

Возврат кода 250 в ответ на команду VRFY после проверки лишь синтаксиса команды (а не указанного адреса), является нарушением этого правила. Естественно, что реализации, «поддерживающие» команду VRFY, которые всегда будут возвращать отклик 550 независимо от корректности адреса, также нарушают это правило.

В публичной сети Internet содержимое списков рассылки стало популярным источником информации об адресах для так называемых спамеров.

Использование команды EXPN для «сбора» адресов вынудило администраторов принять меры против недопустимого использования списков. Однако команды VRFY и EXPN остаются полезными инструментами для аутентифицированных пользователей и внутри административных доменов. Сайты, поддерживающие аутентификацию SMTP, могут выбрать вариант предоставления доступа к командам VRFY и EXPN только аутентифицированных пользователей. Разработчикам **следует** поддерживать команду EXPN, а сайтам **следует** быть осторожными при оценке возможности утечки информации.

Запрет команды VRFY обеспечивает какое-либо повышение уровня безопасности в зависимости от ряда других условий. Во многих случаях ту же информацию о валидности адресов можно получить и с помощью команд RCPT. С другой стороны, особенно для случаев, когда проверка корректности адреса для команд RCPT откладывается до момента получения команды DATA, использование RCPT может не дать никакой информации, тогда как VRFY с высокой вероятностью обеспечит проверку корректности адресов до генерации отклика (см. выше).

7.4. Ремаршрутизация почты на основе откликов 251 и 551

До того, как клиент использует отклики 251 или 551 на команду RCPT для автоматического обновления своего поведения в будущем (корректировка адресной книги пользователя), ему следует проверить подлинность передавшего отклик сервера. Отказ от такой проверки оставляет возможность для организации атак с участием человека (MITM[3]).

[1]Pretty Good Privacy.
[2]Secure/Multipurpose Internet Mail Extensions.
[3]a man in the middle - «человек посередине».

7.5. Разглашение информации в анонсах

Не прекращаются дебаты о преимуществах (отладка) и недостатках (раскрытие информации) анонсирования в откликах на команду HELP информации о типе сервера и номере версии (а в некоторых случаях и доменного имени). Полезность отладочной информации не вызывает сомнений. Те, кто выступает за сохранение её доступности, говорят, что лучше будет сделать серверы SMTP более защищёнными, чем надеяться на то, что сокрытие информации будет повышать уровень защиты. Сайтам рекомендуется принимать эту проблему во внимание, а разработчикам **следует** обеспечивать для серверов возможность предоставления информации о типе и номере версии другим хостам.

7.6. Разглашение информации в полях трассировки

В некоторых обстоятельствах (например, при доставке почты в локальной сети, хосты которой не подключены к Internet напрямую), поля трассировки (Received), вносимые в соответствии с данной спецификацией, могут содержать имена хостов и другую информацию, которую не следует разглашать. Обычно это не создаёт проблем, но для сайтов с высокими требованиями к вопросу разглашения имён это может иметь важное значение. Дополнительные операторы FOR также следует использовать с осторожностью или не использовать совсем в тех случаях, когда многочисленные получатели могут непреднамеренно передать информацию о скрытых получателях (bc) другим.

7.7. Разглашение информации при пересылке сообщений

Как обсуждалось в параграфе 3.4, использование откликов 251 или 551 для идентификации замены адреса, связанного с почтовым ящиком, может приводить к неумышленному разглашению информации. Сайты, для которых эти вопросы играют важную роль, должны соответствующим образом задавать конфигурацию своих серверов.

7.8. Сопротивление атакам

В последние годы наблюдается рост числа атак на серверы SMTP в форме попыток получения адресов для проведения несанкционированных рассылок или попыток срыва работы почтовой службы (например, атаки на службы прикладного уровня). Хотя изучение таких атак выходит за пределы настоящего стандарта, для обеспечения нормального функционирования от серверов требуется возможность детектирования таких атак и принятия мер по своей защите. Например, если сервер обнаруживает передачу большого числа команд RCPT TO, по большей части или полностью включающих некорректные адреса, разумным поведением для сервера будет завершение соединения с генерацией соответствующего количества откликов 5yz (обычно 550).

7.9. Свобода действий сервера SMTP

Несомненно, что сервер SMTP может отвергать почту по любым эксплуатационным или техническим причинам, связанным с сайтом сервера. Однако только кооперация между сайтами и инсталляциями делает возможным функционирование Internet. Если сайты будут слишком активно использовать право отказа от приёма трафика, возможность доставки почты (одна из важных функций Internet) существенно понизится, поэтому следует осторожно и взвешенно принимать решения в части восприятия (отказа) и обработки трафика.

В последние годы использование функций трансляции на случайных сайтах стало применяться как способ сокрытия истинного происхождения почты. Некоторые сайты в результате стали предоставлять функции трансляции только известным или идентифицируемым отправителям и разработчикам **следует** обеспечивать в программах возможность такой фильтрации. Когда почта отвергается по тем или иным причинам, определяемым политикой сайта, **следует** использовать код 550 в откликах на команды EHLO (или HELO), MAIL или RCPT.

8. Регистрация в IANA

Агентство IANA поддерживает три реестра, связанных с данной спецификацией, каждый из которых был открыт для RFC 2821 или раньше. В этом документе расширен третий реестр, как описано ниже. Ссылки на реестры даны на момент публикации документа; IANA не гарантирует постоянную корректность указанных URL. Реестры включают:

- Первый реестр (Simple Mail Transfer Protocol (SMTP) Service Extensions [46]) включает сервисные расширения SMTP и связанные с ними ключевые слова, а также (при необходимости) команды и их параметры. Как сказано в параграфе 2.2.2, ни одна из записей этого реестра не может начинаться с X. Записи могут создаваться только для расширений сервиса (и связанных с ними ключевых слов, параметров и команд), которые определены в стандартах или экспериментальных RFC, одобренных IESG для таких целей.

- Второй реестр (Address Literal Tags [47]) содержит теги, идентифицирующие «дословные» формы записи доменных имён, отличные от адресов IPv4 (эта форма включена в RFC 821 и настоящую спецификацию). Первая запись этого реестра относится к IPv6 (включена в эту спецификацию). Для использования дополнительных вариантов «дословного» представления требуется стандартизация, которой в настоящее время нет ни для одного из них.

- Третий реестр (Mail Transmission Types [46]), основанный RFC 821 и обновленный данной спецификацией, содержит идентификаторы каналов и протоколов, которые могут использоваться в субоператорах via и with трассировочных строк (заголовки Received:), описанных в параграфе 4.4. Идентификаторы каналов и протоколов в дополнение к указанным в этой спецификации могут регистрироваться только путём стандартизации или через экспериментальные расширения протоколов (RFC, одобренные IESG). Размер этого пространства имён для идентификации не ограничен, IESG поддерживает расширение пространства с учётом чёткости документирования и отличия методов, нежели на основе предпочтительности самих методов.
  Добавлены подразделы VIA link types (типы каналов VIA) и WITH protocol types (типы протоколов WITH) для регистрации дополнительных пунктов (Additional-registered-clauses) в соответствии с описанным выше. Реестр будет включать имя пункта (clause), его описание, краткое описание синтаксиса связанной с пунктом строки, и ссылку на источник. По мере определения новых пунктов они могут, в принципе, задавать создание своих реестров, если значения String содержат зарезервированные или ключевые слова. Как идентификаторы каналов и протоколов, дополнительные пункты могут регистрироваться только путём стандартизации или разработки экспериментального протокола, документированного в RFC и одобренного IESG. Пространство имён дополнительных пунктов предназначено для идентификации и не ограничено по размеру. ESG

поддерживает расширение пространства с учётом чёткости документирования пунктов, реального применения или серьёзных предпосылок для использования и отличия пунктов, нежели на основе предпочтительности свойств самих пунктов.

В дополнение к сказанному отметим, что при создании новых трассировочных полей заголовков (т. е. в дополнение к Return-path и Received) эти поля **должны** добавляться в реестр IANA, созданный BCP 90 (RFC 3864) [11] для использования с RFC 5322 [4].

9. Благодарности

Множество людей внесло вклад в подготовку RFC 2821 и отмечено в том документе. В настоящем документе редактор и сообщество должны поблагодарить Dawn Mann и Tony Hansen за помощь в мучительном процессе редактирования и преобразования документа из одного формата в другой.

Ни данный документ, ни RFC 2821 не увидели бы света без участия и проницательности Jon Postel. Его вклад включает и исходную спецификацию SMTP в RFC 821. Значительная часть текста RFC 821 сохранена в настоящем документе, наряду с исходными примерами, которые были слегка обновлены с учётом изменений в спецификации.

Многие люди внесли свои комментарии и предложения через списки рассылок или в письмах автору. Важные поправки и разъяснения предложили множество людей, включая Matti Aarnio, Glenn Anderson, Derek J. Balling, Alex van den Bogaerdt, Stephane Bortzmeyer, Vint Cerf, Jutta Degener, Steve Dorner, Lisa Dusseault, Frank Ellerman, Ned Freed, Randy Gellens, Sabahattin Gucukoglu, Philip Guenther, Arnt Gulbrandsen, Eric Hall, Richard O. Hammer, Tony Hansen, Peter J. Holzer, Kari Hurtta, Bryon Roche Kain, Valdis Kletnieks, Mathias Koerber, John Leslie, Bruce Lilly, Jeff Macdonald, Mark E. Mallett, Mark Martinec, S. Moonesamy, Lyndon Nerenberg, Chris Newman, Douglas Otis, Pete Resnick, Robert A. Rosenberg, Vince Sabio, Hector Santos, David F. Skoll, Paul Smith и Brett Watson.

Работа директоров направления - Lisa Dusseault, Ted Hardie и Chris Newman - помогла начать и завершить подготовку этой спецификации, выполнению этой задачи помог и специальный комитет - все они заслуживают благодарности. Членами этого комитета были (в алфавитном порядке) Dave Crocker, Cyrus Daboo, Tony Finch, Ned Freed, Randall Gellens, Tony Hansen, автор документа и Alexey Melnikov. Tony Hansen также выполнял функции руководства списками рассылок при рассмотрении этого документа, без его усилий не удалось бы беспристрастный и сбалансированный учёт всех предложений; его терпеливость заслуживает отдельной благодарности.

10. Литература

10.1. Нормативные документы

[1] Postel, J., "Simple Mail Transfer Protocol", STD 10, RFC 821, August 1982.

[2] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[3] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, October 1989.

[4] Resnick, P., "Internet Message Format", RFC 5322, October 2008.

[5] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[6] American National Standards Institute (formerly United States of America Standards Institute), "USA Code for Information Interchange", ANSI X3.4-1968, 1968.

[8] ANSI X3.4-1968 has been replaced by newer versions with slight modifications, but the 1968 version remains definitive for the Internet.

[7] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[8] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[9] Newman, C., "ESMTP and LMTP Transmission Types Registration", RFC 3848, July 2004.

[10] Klensin, J., Freed, N., and K. Moore, "SMTP Service Extension for Message Size Declaration", STD 10, RFC 1870, November 1995.

[11] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, September 2004.

10.2. Дополнительная литература

[12] Partridge, C., "Mail routing and the domain system", RFC 974, January 1986.

[13] Klensin, J., Freed, N., Rose, M., Stefferud, E., and D. Crocker, "SMTP Service Extensions", STD 10, RFC 1869, November 1995.

[14] Klensin, J., "Simple Mail Transfer Protocol", RFC 2821, April 2001.

[15] Butler, M., Postel, J., Chase, D., Goldberger, J., and J. Reynolds, "Post Office Protocol: Version 2", RFC 937, February 1985.

[16] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, May 1996.

[17] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.

[18] Gellens, R. and J. Klensin, "Message Submission for Mail", RFC 4409, April 2006.

[19] Freed, N., "SMTP Service Extension for Command Pipelining", STD 60, RFC 2920, September 2000.

[20] Vaudreuil, G., "SMTP Service Extensions for Transmission of Large and Binary MIME Messages", RFC 3030, December 2000.

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

[22] Klensin, J., Freed, N., Rose, M., Stefferud, E., and D. Crocker, "SMTP Service Extension for 8bit-MIMEtransport", RFC 1652, July 1994.

[23] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.

[24] Freed, N. and K. Moore, "MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations", RFC 2231, November 1997.

[25] Vaudreuil, G., "Enhanced Mail System Status Codes", RFC 3463, January 2003.

[26] Hansen, T. and J. Klensin, "A Registry for SMTP Enhanced Mail System Status Codes", BCP 138, RFC 5248, June 2008.

[27] Freed, N., "Behavior of and Requirements for Internet Firewalls", RFC 2979, October 2000.

[28] Crocker, D., "Standard for the format of ARPA Internet text messages", STD 11, RFC 822, August 1982.

[29] Wong, M. and W. Schlitt, "Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1", RFC 4408, April 2006.

[30] Fenton, J., "Analysis of Threats Motivating DomainKeys Identified Mail (DKIM)", RFC 4686, September 2006.

[31] Allman, E., Callas, J., Delany, M., Libbey, M., Fenton, J., and M. Thomas, "DomainKeys Identified Mail (DKIM) Signatures", RFC 4871, May 2007.

[32] Moore, K., "Simple Mail Transfer Protocol (SMTP) Service Extension for Delivery Status Notifications (DSNs)", RFC 3461, January 2003.

[33] Moore, K. and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", RFC 3464, January 2003.

[34] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, October 1985.

[35] Kille, S., "MIXER (Mime Internet X.400 Enhanced Relay): Mapping between X.400 and RFC 822/MIME", RFC 2156, January 1998.

[36] De Winter, J., "SMTP Service Extension for Remote Message Queue Starting", RFC 1985, August 1996.

[37] Hansen, T. and G. Vaudreuil, "Message Disposition Notification", RFC 3798, May 2004.

[38] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, July 1997.

[39] Nakamura, M. and J. Hagino, "SMTP Operational Experience in Mixed IPv4/v6 Environments", RFC 3974, January 2005.

[40] Partridge, C., "Duplicate messages and SMTP", RFC 1047, February 1988.

[41] Crispin, M., "Interactive Mail Access Protocol: Version 2", RFC 1176, August 1990.

[42] Lambert, M., "PCMAIL: A distributed mail system for personal computers", RFC 1056, June 1988.

[43] Galvin, J., Murphy, S., Crocker, S., and N. Freed, "Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted", RFC 1847, October 1995.

[44] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, November 2007.

[45] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, July 2004.

[46] Internet Assigned Number Authority (IANA), "IANA Mail Parameters", 2007, <http://www.iana.org/assignments/mail-parameters>.

[47] Internet Assigned Number Authority (IANA), "Address Literal Tags", 2007, <http://www.iana.org/assignments/address-literal-tags>.

Приложение A. Транспортный сервис TCP

Соединения TCP поддерживают передачу 8-битовых байтов, а данные SMTP представляют собой 7-битовые символы ASCII. Каждый символ передаётся в 8-битовом байте с нулевым значением старшего бита. Сервисные расширения могут изменять это правило и разрешать передачу 8-битовых байтов для содержимого сообщений или, если приняты специальные меры для этого, команд и откликов SMTP.

Приложение B. Генерация команд SMTP из полей заголовка RFC 822

Некоторые системы используют раздел заголовков (и только его) RFC 822 в протоколах представления почты, а в остальных случаях генерируют команды SMTP на основе заголовков RFC 822, когда такие сообщения передаются АДП (MTA) от агентов ППА (MUA). Поскольку протокол взаимодействия АДП – ППА является частным и не задается стандартами Internet, в таких случаях могут возникать проблемы. Например, повторяющиеся проблемы могут возникать при обработке копий bcc и перераспределении списков, когда информация, потенциально относящаяся к почтовому конверту, не отделяется в процессе обработки от информации из заголовков и не хранится отдельно от неё.

Агентам ППА рекомендуется предоставлять своему первому АДП (submission client) конверт отдельно от сообщения. Однако, если конверты не поддерживаются, **следует** генерировать команды SMTP, используя приведённые правила:

1. Каждый адрес получателя из полей заголовка TO, CC, BCC **следует** копировать в команду RCPT (генерируя, при необходимости, нужное число копий сообщения для помещения в очередь или доставки). Сюда включаются все адреса, перечисленные в «группе» RFC 822. Все поля BCC **следует** удалять из заголовков. После завершения

такой обработки оставшиеся поля заголовков **следует** проверить, чтобы убедиться, что осталось хотя бы одно поле TO, CC или BCC. При отсутствии **следует** поместить в заголовок поле BCC без дополнительной информации, как указано в работе [4].

2. Адрес возврата в команде MAIL **следует** (по возможности) получать из системной идентификации представляющего почту (локального) пользователя или из поля From:. При доступности системной идентификации, эти данные **следует** также копировать в поле заголовка Sender, если информация отличается от адреса в поле From (все имеющиеся поля Sender следует удалить). Система может позволять представляющим почту пользователям переписывать адрес возврата в конверте, но можно ограничить доступ к этому только привилегированными пользователями. Это не предотвращает подмены почтовых адресов, но осложняет такую подмену (см. параграф 7.1).

При таком использовании агентов АДП они несут ответственность за корректность передаваемого сообщения. Механизм проверки корректности и обработка (или возврат) сообщений, некорректность которых обнаружена по прибытии, являются частью интерфейса АДП – ППА (MUA-MTA) и не рассматриваются в данной спецификации.

Протокол представления почты, основанный только на стандарте RFC 822, **недопустимо** использовать на шлюзах из других (не SMTP) почтовых систем в среду SMTP. Дополнительные данные для конструирования заголовков требуется получать из некоторых источников в другой среде (дополнительные заголовки или конверт).

Попытки передавать сообщения через шлюзы, используя только поля заголовка To и Cc будут приводить к возникновению почтовых петель и другим нарушениям в работе почтовой системы Internet. Эти проблемы будут возникать особенно часто в случаях отправки сообщений через списки рассылки Internet и при распределении почты в чужие среды с использованием информации из конверта. Когда при пересылке таких сообщений учитываются только заголовки, возникновение почтовых петель обратно в среду Internet (и на почтовые списки) почти неизбежно.

Приложение C. Маршруты Source Route

Исторически поле <reverse-path> содержало в source routing список промежуточных хостов и имя почтового ящика отправителя. Исторически, первым в списке <reverse-path> указывался хост, подавший команду MAIL; сегодня **не следует** использовать маршруты source route в обратном пути. Подобно этому поле <forward-path> может быть списком хостов source routing и адреса получателя. Однако, в общем случае, в поле <forward-path> **следует** включать только почтовый ящик и доменное имя получателя, отдавая решение задачи маршрутизации почты на откуп системе DNS. Использование явных маршрутов осуждается (см. Приложение F.2) - хотя серверы и **должны** быть готовы к получению и обработке таких маршрутов (см. 3.3 и Приложение F.2), клиентам **не следует** передавать явные маршруты и этот параграф включён в спецификацию только для обеспечения преемственности. Данная спецификация несколько отличается от RFC 821 для предотвращения действий серверов, приводящих к путанице последующие серверы и клиентов, которые не ожидают полной реализации source route.

Для целей трансляции прямой путь может быть маршрутом source route в форме @ONE,@TWO:JOE@THREE, где ONE, TWO, THREE **должны** быть полными доменными именами. Такая форма используется для того, чтобы можно было отличить адреса от маршрутов. Почтовый ящик (в данном случае, JOE@THREE) представляет собой абсолютный адрес, а маршрут - информацию для доставки. Эти понятия не следует путать.

При использовании source route требования RFC 821 и приведённый ниже текст вступают в противоречие в части механизма создания и обновления прямого пути. Сервер SMTP, достигнутый по маршруту source route (например, его доменное имя появляется первым в списке прямого пути), **должен** удалить своё доменное имя из прямых путей, где это имя появляется, до пересылки сообщения и **может** удалить всю остальную информацию source route. В соответствии с данной спецификацией серверу **не следует** менять обратный путь.

Отметим, что прямой и обратный пути появляются в командах и откликах SMTP, но не являются необходимыми в сообщении (т. е., нет необходимости включения таких путей и особенно описанного здесь синтаксиса в поля To:, From:, CC: и т. п.). И наоборот, для серверов SMTP **недопустимо** получать информацию на основе этих полей при окончательной доставке сообщения.

Когда, несмотря на приведённые выше рекомендации, список хостов присутствует, этот список является «обратным» маршрутом source route и показывает, что почта транслировалась через каждый хост в списке (первым в списке указан последний по времени транслятор). Этот список используется как source route для возврата отправителю уведомлений о невозможности доставки. Если в нарушение приведённых здесь рекомендаций хост-транслятор добавляет себя в начало списка, он **должен** использовать имя, которое известно в транспортной среде, куда транслируется почта, а не в среде, откуда почта поступила (если эти имена различаются). Отметим, что такие ситуации могут с лёгкостью возникать из случаев, когда некоторые транслирующие хосты добавляют свои имена в обратный путь source route, а другие не делают этого, создавая разрывы в маршрутном списке. Это другая причина, по которой серверам при необходимости возврата сообщения **следует** полностью игнорировать source route и просто использовать домен, указанный в Mailbox.

Приложение D. Сценарии

В этом приложении приведено несколько примеров полных сценариев сеансов SMTP. Знаком C: обозначается отправитель (клиент SMTP), а знаком S: - сервер SMTP.

D.1. Сценарий типовой транзакции SMTP

Рассматриваемый ниже пример показывает передачу сообщения, отправленного Смитом (Smith) с хоста bar.com адресатам Jones, Green, Brown на foo.com. Предполагается, что bar.com контактирует с foo.com напрямую. Почта для Jones и Brown принимается, а Green не имеет почтового ящика на foo.com.

```
S: 220 foo.com Simple Mail Transfer Service Ready
C: EHLO bar.com
S: 250-foo.com greets bar.com
S: 250-8BITMIME
S: 250-SIZE
```

```
S: 250-DSN
S: 250 HELP
C: MAIL FROM:<Smith@bar.com>
S: 250 OK
C: RCPT TO:<Jones@foo.com>
S: 250 OK
C: RCPT TO:<Green@foo.com>
S: 550 No such user here
C: RCPT TO:<Brown@foo.com>
S: 250 OK
C: DATA
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: Blah blah blah...
C: ...etc. etc. etc.
C: .
S: 250 OK
C: QUIT
S: 221 foo.com Service closing transmission channel
```

D.2. Сценарий прерванной транзакции SMTP

```
S: 220 foo.com Simple Mail Transfer Service Ready
C: EHLO bar.com
S: 250-foo.com greets bar.com
S: 250-8BITMIME
S: 250-SIZE
S: 250-DSN
S: 250 HELP
C: MAIL FROM:<Smith@bar.com>
S: 250 OK
C: RCPT TO:<Jones@foo.com>
S: 250 OK
C: RCPT TO:<Green@foo.com>
S: 550 No such user here
C: RSET
S: 250 OK
C: QUIT
S: 221 foo.com Service closing transmission channel
```

D.3. Сценарий с трансляцией

Этап 1 - Отправитель → транслятор.

Хост-отправитель запрашивает у DNS данные для XYZ.COM (адрес получателя) и получает записи DNS MX, указывающие xyz.com, как наиболее предпочтительный хост, а foo.com - как наименее предпочтительный. Отправитель пытается соединиться с xyz.com, но терпит неудачу. Тогда он организует соединение с foo.com, сценарий которого показан ниже:

```
S: 220 foo.com Simple Mail Transfer Service Ready
C: EHLO bar.com
S: 250-foo.com greets bar.com
S: 250-8BITMIME
S: 250-SIZE
S: 250-DSN
S: 250 HELP
C: MAIL FROM:<JQP@bar.com>
S: 250 OK
C: RCPT TO:<Jones@XYZ.COM>
S: 250 OK
C: DATA
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: Date: Thu, 21 May 1998 05:33:29 -0700
C: From: John Q. Public <JQP@bar.com>
C: Subject: The Next Meeting of the Board
C: To: Jones@xyz.com
C:
C: Bill:
C: The next meeting of the board of directors will be
C: on Tuesday.
C: John.
C: .
S: 250 OK
C: QUIT
S: 221 foo.com Service closing transmission channel
```

Этап 2 - Транслятор → хост получателя

Хост foo.com, получивший сообщение, запрашивает у DNS данные для xyz.com. Он получает такой же набор записей MX, но не может использовать ни свой адрес, ни любой адрес из списка, который менее предпочтителен, по сравнению с xyz.com. Тогда транслятор организует соединение с xyz.com:

```
S: 220 xyz.com Simple Mail Transfer Service Ready
C: EHLO foo.com
S: 250 xyz.com is on the air
C: MAIL FROM:<JQP@bar.com>
S: 250 OK
C: RCPT TO:<Jones@XYZ.COM>
S: 250 OK
C: DATA
```

```
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: Received: from bar.com by foo.com ; Thu, 21 May 1998
C:     05:33:29 -0700
C: Date: Thu, 21 May 1998 05:33:22 -0700
C: From: John Q. Public <JQP@bar.com>
C: Subject:  The Next Meeting of the Board
C: To: Jones@xyz.com
C:
C: Bill:
C: The next meeting of the board of directors will be
C: on Tuesday.
C:                          John.
C: .
S: 250 OK
C: QUIT
S: 221 foo.com Service closing transmission channel
```

D.4. Сценарий проверки и передачи

```
S: 220 foo.com Simple Mail Transfer Service Ready
C: EHLO bar.com
S: 250-foo.com greets bar.com
S: 250-8BITMIME
S: 250-SIZE
S: 250-DSN
S: 250-VRFY
S: 250 HELP
C: VRFY Crispin
S: 250 Mark Crispin <Admin.MRC@foo.com>
C: MAIL FROM:<EAK@bar.com>
S: 250 OK
C: RCPT TO:<Admin.MRC@foo.com>
S: 250 OK
C: DATA
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: Blah blah blah...
C: ...etc. etc. etc.
C: .
S: 250 OK
C: QUIT
S: 221 foo.com Service closing transmission channel
```

Приложение E. Другие вопросы, связанные со шлюзами

В общем случае, шлюзам между Internet и другими почтовыми системами **следует** пытаться сохранить семантику при передаче сообщения через границу между двумя почтовыми системами. Шлюзы, которые пытаются сделать «вырезки» путём отображения (например, передача информации из конверта одной среды в заголовок или тело сообщения в другой среде), в общем случае не могут обеспечить требуемого уровня передачи информации. Системы, транслирующие между средами, которые не могут поддерживать одновременно конверты и заголовки почты Internet, должны понимать, что в таких случаях потеря некоторой информации практически неизбежна.

Приложение F. Отменённые возможности RFC 821

Некоторые возможности RFC 821 признаны проблематичными и их **не следует** использовать в почте Internet.

F.1. Команда TURN

Эта команда, описанная в RFC 821, затрагивает важные аспекты безопасности, поскольку в отсутствие жёсткой аутентификации для хостов, запрашивающих смену ролей клиента и сервера, такую команду можно с лёгкостью использовать для переадресации почты. Использование этой команды осуждается и системам SMTP **не следует** применять её без аутентификации клиента сервером.

F.2. Задаваемая отправителем маршрутизация

RFC 821 использует концепцию явного задания маршрута отправителем для доставки почты с одного хоста на другой через промежуточные трансляторы. Необходимость использования такой маршрутизации в обычной почте отпала после появления в DNS записей MX. Существенный вклад в отказ от такой маршрутизации внёс документ RFC 1123, в соответствии с которым после символа @ в адресе должно указываться полное доменное имя (FQDN). Следовательно, единственной причиной поддержки source route является взаимодействие со старыми клиентами SMTP или агентами MUA, а также отладка почтовых систем. Однако такая маршрутизация может быть полезна при возникновении серьёзных проблем временного характера (типа релевантности записей DNS).

Серверы SMTP **должны** продолжать восприятие синтаксиса source route в соответствии с данной спецификацией и RFC 1123. При необходимости серверы **могут** игнорировать явные маршруты, используя из адреса только доменное имя. При использовании source route сообщение **должно** пересылаться в первый указанный в адресе домен. В частности, для серверов **недопустимо** сокращение маршрута source route.

Клиентам **не следует** использовать явную маршрутизацию за исключением нештатных ситуаций типа отладки, потенциальной трансляции в обход брандмауэров или случаев возникновения конфигурационных ошибок.

F.3. Команда HELO

Как было указано в параграфах 3.1 и 4.1.1, **следует** отдавать предпочтение команде EHLO, нежели устаревшей команде HELO. Серверы **должны** принимать и обрабатывать команды HELO для поддержки старых клиентов.

F.4. #-литералы

В RFC 821 указана возможность задания адресов Internet с помощью десятичного представления номера хоста с префиксом #. На практике с появлением TCP/IP актуальность такого представления была утрачена. В настоящее время этот вариант осуждается и **недопустим** для использования.

F.5. Даты и годы

При включении клиентами и серверами SMTP значений даты в сообщения (например, в поля трассировки) **должно** использоваться 4-значное представление года. Двухзначное представление осуждается, а трехзначное никогда не допускалось в почтовых системах Internet.

F.6. Дополнительные команды прямой передачи

В дополнение к спецификации механизма доставки сообщений в почтовые ящики пользователей, RFC 821 обеспечивает добавочные команды для прямой доставки сообщений на консоль пользователя. Эти команды (SEND, SAML, SOML) использовались в реализациях достаточно редко, а изменения в технологии рабочих станций и появление других протоколов могут привести к полному забвению этих команд даже при их поддержке в программах.

Клиентам **не следует** предоставлять услуги SEND, SAML или SOML, но серверы их **могут** реализовать. При реализации этих служб сервером **должна** использоваться модель, приведённая в спецификации RFC 821, а имена команд **должны** публиковаться в отклике на команду EHLO.

Адрес автора

John C. Klensin

1770 Massachusetts Ave, Suite 322

Cambridge, MA 02140

USA

EMail: john+smtp@jck.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru