

# Лабораторная работа № 10

## Передача почтовых сообщений. Протокол SMTP.

### *Краткие теоретические сведения*

#### 1. Принципы организации электронной почты в Internet.

Электронная почта – один из важнейших информационных ресурсов Internet, самое массовое средство электронных коммуникаций. Любой из пользователей Internet может получить свой собственный почтовый ящик в сети.

Самым простым решением по организации обмена сообщениями напрашивается следующее. Допустим, что пользователь А имеет подключенный к некоторой сети передачи данных (СПД) Узел А. И пользователь Б с Узлом Б является абонентом той же сети. Тогда для передачи сообщений от А к Б необходимо организовать протокол обмена сообщениями (или просто, почтой) и этот протокол должен быть реализован на обоих узлах. Предположим, что СПД не предоставляет напрямую такого протокола, тогда на А и Б ложится вся нагрузка по приему/передаче почты. А что если пользователь является абонентом нескольких сетей? Тогда на его компьютере должно будет установлено все необходимое программное обеспечение, реализующее всевозможные протоколы. Нагрузка возрастет, если помимо программных интерфейсов необходимо будет устанавливать аппаратные.

Действительные проблемы возникают, когда оказывается, что удаленный узел (Б) недоступен по ряду причин:

- он мог сменить имя узла и не уведомил об этом корреспондента;
- возможно его узел просто отключен или просто на данный момент не существует прямого маршрута от А к Б.

Во всех этих случаях почта не дойдет до узла Б. В лучшем случае пользователь А получит уведомление о невозможности достичь Узел Б, а в худшем – сообщение просто затеряется в СПД.

Таким образом, для надежной доставки сообщений был разработан подход, в котором Электронная почта во многом похожа на обычную почтовую службу. Корреспонденция подготавливается пользователем на своем рабочем месте либо программой подготовки почты, либо просто обычным текстовым редактором. Затем пользователь должен вызвать программу отправки почты (как правило, программа подготовки почты вызывает программу отправки автоматически). Стандартной программой отправки на Unix системах является программа Sendmail. Sendmail работает как почтовый курьер, который доставляет обычную почту в отделение связи для дальнейшей рассылки. В Unix-системах Sendmail сама является отделением связи. Она сортирует почту и рассылает ее адресатам. Для пользователей персональных компьютеров, имеющих почтовые ящики на своих машинах и работающих с почтовыми серверами через коммутируемые телефонные линии, могут потребоваться дополнительные действия. Так, например, пользователи почтовой службы Relcom должны запускать программу **uucpс**, которая осуществляет доставку почты на почтовый сервер.

Для передачи электронной почты в Internet разработан специальный протокол **Simple Mail Transfer Protocol (SMTP)**, который является протоколом прикладного уровня и использует транспортный протокол **TCP**.

#### 2. Форматы почтовых адресов.

Основой любой почтовой службы является система адресов. Без точного адреса невозможно доставить почту адресату. В Internet принята система адресов, которая базируется на доменном адресе машины, подключенной к сети. Например, для пользователя paul машины с адресом polyn.net.kiae.su почтовый адрес будет выглядеть как:

paul@polyn.net.kiae.su

Таким образом, адрес состоит из двух частей: идентификатора пользователя, который записывается перед знаком "коммерческого эт" – "@", и доменного адреса машины, который записывается после знака "@". Такой тип адреса называется маршрутонезависимым (routeless). Т.е. в адресе нет прямого указания, через какие машины будет лежать маршрут почтового сообщения. Противоположным ему является адрес UUCP, который для нашего примера выглядит так:

net!kiae!su!polyn!paul

Здесь явно указываются промежуточные узлы. Маршрут доставки от отправителя до пользователя paul машины polyn будет следующим:

-> net -> kiae -> su -> polyn

Программа рассылки почты Sendmail сама преобразует адреса формата Internet в адреса формата UUCP, если доставка сообщения осуществляется по этому протоколу.

Такой принцип доставки здесь был назван **STOP-GO**. Такое название обусловлено тем, что каждый промежуточный узел вносит ощутимую задержку к общему времени передачи. Дело в том, что UUCP-соединение – это связка типа точка-точка, поэтому накладывается ограничение на число подключенных машин к передающему узлу в один момент времени. Также существуют временные ограничения, связанные с моментом соединения, его продолжительностью и т.д. Поэтому нельзя быть уверенным в том, что промежуточный узел передаст почту, как только она будет доставлена к нему. На это может уйти даже несколько суток, если почтовый сервер загружен другими

сообщениями. Далее мы не будем рассматривать почту UUCP, т.к. она устарела, хотя до сих пор применяется как самый дешевый способ доставки почты для организаций и частных лиц.

### 3. Компоненты электронной почты.

На рис. 1 приведены компоненты системы электронной почты в рамках сети Internet.

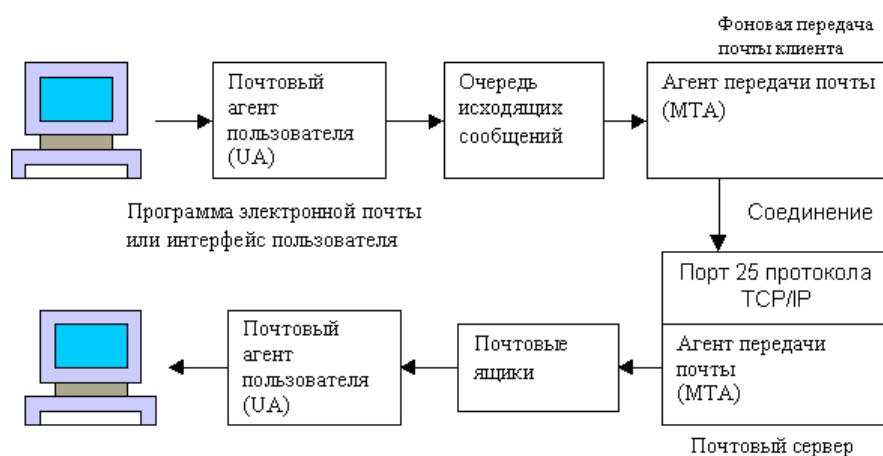


Рисунок 1. Компоненты почтовой системы Internet.

Термин "агент" довольно часто встречается в документации Internet. "Агент" – это программа специального назначения, выполняющая действия для пользователя или другой программы. В большинстве случаев почтовая программа называется агентом пользователя (User Agent, UA). Точно так же агент передачи почты (Message Transfer Agent, MTA) представляет собой клиент или сервер, выполняющий задачи по доставке или получению почты на сетевом компьютере.

В принципе, пользовательский агент отделен от агента передачи почты. Конечно, их можно объединить в одной программе, но все равно это будут отдельные логические модули. Будучи взаимосвязаны, оба агента выполняют совершенно различные функции. Пользователи системы Unix хорошо знакомы с такими программами, как **MH**, **Mail**, **Elm**, **Mush** и **Pine**. Для пользователей Windows самой известной программой, возможно, является **Microsoft Outlook Express**. Все эти программы – агенты пользователя. Каждая обеспечивает интерфейс между пользователем и системой электронной почты Интернет. Большинство современных программ позволяют отправлять почту с помощью **SMTP**, принимать почту по **POP3** и **IMAP4** протоколам, а так же читать группы новостей (**NNTP**).

Система электронной почты представлена агентами передачи почты, MTA. До того как обсудить задачи пользовательского агента, необходимо узнать немного больше о том, что же такое MTA. MTA умеют устанавливать соединение для связи с другими MTA. Протоколом этого соединения, как правило, является простой протокол передачи почты (Simple Message Transfer Protocol, SMTP).

Агент передачи почты – основной компонент системы передачи почты Internet. Как уже говорилось, MTA как бы представляет данный сетевой компьютер для сетевой системы электронной почты. Пользователи редко имеют дело с MTA, поскольку он не вполне "дружелюбен", однако без него не обходится ни одна почтовая система. После того как UA pošлет сообщение в выходную очередь, за дело принимается MTA. Он извлекает сообщение и посылает его другому MTA. Этот процесс продолжается до тех пор, пока сообщение не достигнет компьютера-получателя. Для передачи сообщений по TCP-соединению большинство MTA пользуются протоколом SMTP. Сообщения форматованы по правилам виртуального сетевого терминала (NVT), то есть в NVT ASCII. NVT подобен виртуальному сетевому протоколу и нужен затем, чтобы скрыть различия в восприятии разными компьютерами разных символов, например переводов каретки, переводов строки, маркеров конца строки, очистки экрана и т. д. Символ в NVT состоит из семи битов набора ASCII и является буквой, цифрой или знаком пунктуации. Семи битный набор ASCII часто называется NVT ASCII.

### 4. Модель протокола.

Взаимодействие в рамках SMTP строится по принципу двусторонней связи, которая устанавливается между отправителем и получателем почтового сообщения. При этом отправитель инициирует соединение и посылает запросы на обслуживание, а получатель – отвечает на эти запросы. Фактически отправитель выступает в роли клиента, а получатель – сервера.

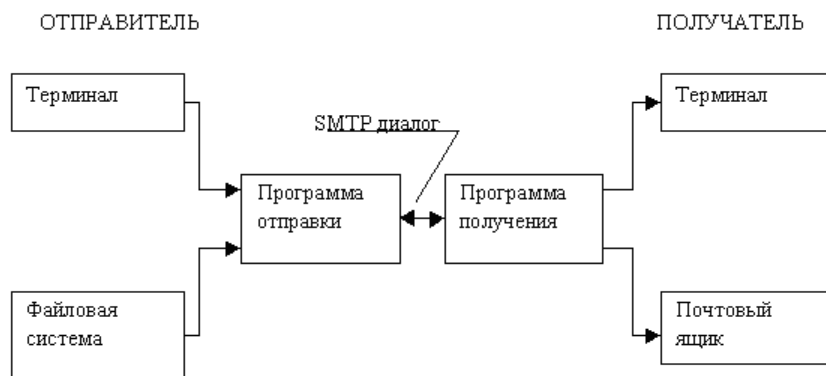


Рисунок 2. Схема взаимодействия по протоколу SMTP.

#### Команды SMTP

Простой протокол передачи почты обеспечивает двухсторонний обмен сообщениями между локальным клиентом и удаленным сервером МТА. МТА-клиент шлет команды МТА-серверу, а он, в свою очередь, отвечает клиенту. Другими словами, протокол SMTP требует получать ответы от приемника команд SMTP. Обмен командами и ответами на них называется почтовой транзакцией (mail transaction). Данные передаются в формате NVT ASCII. Кроме того, команды тоже передаются в формате NVT ASCII. Команды передаются в форме ключевых слов, а не специальных символов, и указывают на необходимость совершить ту или иную операцию. В табл.1 приведен список ключевых слов (команд), определенный в спецификации SMTP – RFC 821.

Таблица 1. Команды простого протокола передачи почты (SMTP)

Команда	Обязательна	Описание
<b>HELO</b>	X	Идентифицирует модуль-передатчик для модуля-приемника (hello).
<b>MAIL</b>	X	Начинает почтовую транзакцию, которая завершается передачей данных в один или несколько почтовых ящиков (mail).
<b>RCPT</b>	X	Идентифицирует получателя почтового сообщения (recipient).
<b>DATA</b>		Строки, следующие за этой командой, рассматриваются получателем как данные почтового сообщения. В случае SMTP, почтовое сообщение заканчивается комбинацией символов: CRLF-точка-CRLF.
<b>RSET</b>		Прерывает текущую почтовую транзакцию (reset).
<b>NOOP</b>		Требует от получателя не предпринимать никаких действий, а только выдать ответ OK. Используется главным образом для тестирования.(No operation).
<b>QUIT</b>		Требует выдать ответ OK и закрыть текущее соединение.
<b>VRFY</b>		Требует от приемника подтвердить, что ее аргумент является действительным именем пользователя.
<b>SEND</b>		Начинает почтовую транзакцию, доставляющую данные на один или несколько терминалов (а не в почтовый ящик).
<b>SOML</b>		Начинает транзакцию MAIL или SEND, доставляющую данные на один или несколько терминалов или в почтовые ящики.
<b>SAML</b>		Начинает транзакцию MAIL и SEND, доставляющие данные на один или несколько терминалов и в почтовые ящики.

<b>EXPN</b>		Команда SMTP-приемнику подтвердить, действительно ли аргумент является адресом почтовой рассылки и если да, вернуть адрес получателя сообщения (expand).
<b>HELP</b>		Команда SMTP-приемнику вернуть сообщение-справку о его командах.
<b>TURN</b>		Команда SMTP-приемнику либо сказать ОК и поменяться ролями, то есть стать SMTP- передатчиком, либо послать сообщение-отказ и остаться в роли SMTP-приемника.

**Примечание:** В RFC 821 сказано, что команда VRFY не является обязательной для минимального набора команд SMTP. Однако в RFC 1123 <Требования для сетевых компьютеров Internet – приложения и обеспечение работы> (Requirements for Internet Hosts – Application and Support, Braden, 1989), команда VRFY фигурирует в списке обязательных для Internet команд реализации SMTP.

В соответствии со спецификацией команды, помеченные крестиком (X) в табл.1, обязаны присутствовать в любой реализации SMTP. Остальные команды SMTP могут быть реализованы дополнительно. Каждая SMTP-команда должна заканчиваться либо пробелом (если у нее есть аргумент), либо комбинацией CRLF. В описании команд употреблялось слово <данные", а не <сообщение>. Этим подчеркивалось, что, кроме текста, SMTP позволяет передавать и двоичную информацию, например графические или звуковые файлы. Другими словами, SMTP способен передавать данные любого содержания, а не только текстовые сообщения.

## 5. Команды SMTP.

SMTP обеспечивает двухстороннюю связь между агентами передачи почты (MTA), клиентом и сервером. Клиенты шлют команды серверу, а серверы отвечают клиентам. Однако SMTP оговаривает последовательность SMTP-команд. Лучший способ понять это – взглянуть на образец почтовой транзакции. Следующий пример (он взят целиком из RFC 821) демонстрирует типичную почтовую транзакцию. В примере фигурирует мистер Smith (на компьютере usc.edu), посылающий сообщения мистерам Jones, Green и Brown (на компьютере mit.edu). Агент передачи почты узла mit.edu принимает почту для мистеров Jones и Brown, однако не знает, где расположен почтовый ящик мистера Green.

Для целей дальнейшего повествования каждой строке присвоен номер и обозначено, кому они принадлежат – передатчику или приемнику. Текст справа от слов <RECEIVER> или <SENDER> содержит действительно передаваемые данные. Трехзначные цифровые комбинации в начале передаваемых строк обозначают коды ответа (их значение объясняется позже). Ответ SMTP похож на сообщения-подтверждения о доставке, поскольку появляется лишь в том случае, когда приемник получил данные.

```

1  RECEIVER      220 mit.edu Simple Mail Transfer Service Ready
2  SENDER        HELO usc.edu
3  RECEIVER      250 mit.edu
4  SENDER        MAIL FROM: <Smith@usc.edu>
5  RECEIVER      250 OK
6  SENDER        RCPT TO:<Jones@mit.edu>
7  RECEIVER      250 OK
8  SENDER        RCPT TO:<Green@mit.edu>
9  RECEIVER      550 No such user here
10 SENDER        RCPT TO:<Brown@mit.edu>
11 RECEIVER      250 OK
12 SENDER        DATA
13 RECEIVER      354 Start mail input; end with <CRLF>.<CRLF>
14 SENDER        Hello...
15 SENDER        etc
16 SENDER        .
17 RECEIVER      250 OK
18 SENDER        QUIT
19 RECEIVER      221 mit.edu Service closing transmission channel

```

Как видно из строки 1, когда SMTP-клиент устанавливает TCP-соединение с портом протокола 25, SMTP-сервер отвечает кодом 220. Это означает, что соединение успешно установлено:

1. RECEIVER: 220 mit.edu Simple Mail Transfer Service Ready

После того как МТА компьютеров mit.edu и usc.edu установили соединение и обменялись приветствием, первой командой, согласно спецификации, должна быть команда **HELO**. Как указано в строке 2, SMTP-клиент передает **HELO**, указывая имя своего компьютера в качестве аргумента. Другими словами, он сообщает: <Привет, я – usc.edu>. Команда **HELO** употребляется с аргументом, как показано ниже:

2. SENDER: HELO usc.edu

В ответ на **HELO** приемник выдает код 250, сообщая передатчику о том, что команда принята и обработана:

3. RECEIVER: 250 mit.edu

После установления TCP-соединения и идентификации (при помощи **HELO**) SMTP-клиент приступает к почтовой транзакции. Для начала он выполняет одну из следующих команд: **MAIL**, **SEND**, **SOML** или **SAML**. В нашем примере использована команда **MAIL**:

4. SENDER: MAIL FROM:<Smith@usc.edu>

Все четыре команды, **MAIL**, **SEND**, **SOML** и **SAML**, имеют одинаковый синтаксис:

**MAIL** <пробел> FROM:<reverse-path> <carriage-return line-feed>

**Примечание:** Команды **SEND**, **SOML** и **SAML** дополнительные и используются довольно редко.

Аргумент <обратный путь> (reverse path) указывает серверу, кому в случае ошибки отослать соответствующее сообщение. Мы еще рассмотрим его подробнее. На данный момент для нас важно, что в аргументе содержится адрес источника сообщения (в нашем случае, Smith@usc.edu). После того как сервер выдал код ответа 250 (строка 5), согласившись обработать сообщение от Smith@usc.edu необходимо указать получателя сообщения. Это делается при помощи команды RCPT. Команда RCPT имеет аргумент – имя получателя. На одну команду приходится только одно имя, поэтому, если получателей несколько, команда **RCPT** выдается несколько раз. В нашем примере команды **RCPT** выполняются в строках 6, 8 и 10. Синтаксис **RCPT** похож на синтаксис команды **MAIL**:

RCPT <пробел> TO:<forward-path> <CRLF>

Однако, в отличие от **MAIL**, аргумент **RCPT** начинается со слова <TO:>. Содержимое аргумента – путь передачи сообщения (forward path), а не обратный путь. На данный момент для нас важно, что в пути передачи сообщения указано имя почтового ящика получателя. Выдав команду **RCPT**, МТА-клиент ожидает получить ответ с кодом 250. Однако в ответ на восьмую строку

8. SENDER: RCPT TO:<Green@mit.edu>

сервер отвечает кодом 550:

9. RECEIVER: 550 No such user here

Код ответа 550 означает, что МТА не в состоянии выполнить запрос клиента, поскольку не знает, как доставить почту указанному пользователю. То есть, скорее всего, у мистера по фамилии Green нет почтового ящика (Green@mit.edu) на этом компьютере. В протоколе SMTP сказано, что сервер обязан информировать клиента об отсутствии почтового ящика получателя сообщения. Однако в спецификации SMTP ничего не говорится о том, как клиент должен реагировать на это сообщение.

После того как посланы все команды **RCPT**, клиент начинает передачу данных при помощи команды **DATA**. В строке 12 показано, как МТА-клиент (передатчик) высылает команду **DATA**, в строке 13 – как сервер отвечает кодом 354. Этот код означает, что передача данных разрешена и должна заканчиваться комбинацией CRLF-<точка>-CRLF (новой строкой, содержащей только точку).

12. SENDER: DATA

13. RECEIVER: 354 Start mail input; end with <CRLF>.<CRLF>

После того как получен код 354, клиент может начать передачу данных. МТА-сервер, в свою очередь, помещает принятые данные в очереди входящих сообщений. Сервер не высылает никаких ответов до тех пор, пока не получит комбинацию CRLF-точка-CRLF от клиента, означающую конец передачи данных. Как показано в строках 16 и 17, в ответ на полученную комбинацию CRLF-<точка>-CRLF, сервер выдает код 250. Как мы уже говорили, код ответа 250 означает успешное окончание операции:

16. SENDER: .

17. RECEIVER: 250 OK

Для того чтобы закончить почтовую транзакцию, клиент, по правилам SMTP, обязан послать команду **QUIT**. Сервер, в свою очередь, отвечает кодом 221. Этот код подтверждает клиенту, что соединение будет закрыто, после чего соединение действительно закрывается:

18. SENDER: QUIT

19. RECEIVER: 221 mit.edu Service closing transmission channel

В любой момент во время транзакции клиент может использовать команды **NOOP**, **HELP**, **EXPN** и **VRFY**. В ответ на каждую команду сервер высылает клиенту определенную информацию. Конечно, в зависимости от ответа клиент может предпринять определенные действия, однако спецификация SMTP ничего не говорит по этому поводу. Например, клиент-МТА может передать команду **VRFY** для того, чтобы убедиться, что имя пользователя действительно. Если сервер ответит, что данного имени не существует, клиент МТА может не передавать почту для этого пользователя. В спецификации SMTP, однако, на этот счет нет никаких указаний – клиент может ничего не делать в ответ на команду **VRFY**. МТА-клиент может ничего не делать также в ответ на команды **NOOP**, **HELP** и **EXPN** – ответственность целиком лежит на разработчике конкретной реализации МТА.

В спецификации SMTP требуется, чтобы сервер отвечал на каждую команду SMTP-клиента. МТА-сервер отвечает трехзначной комбинацией цифр, называемой кодом ответа. Вместе с кодом ответа, как правило, передается одна или несколько строк текстовой информации.

**Примечание:** Несколько строк текста, как правило, сопровождают только команды **EXPN** и **HELP**. В спецификации SMTP, однако, ответ на любую команду может состоять из нескольких строк текста.

Каждая цифра в коде ответа имеет определенный смысл. Первая цифра означает, было ли выполнение команды успешно (2), неуспешно (5) или еще не закончилось (3). Как указано в приложении Е документа RFC 821, простой SMTP-клиент может анализировать только первую цифру в ответе сервера, и на основании ее продолжать свои действия. Вторая и третья цифры кода ответа разъясняют значение первой. Если вы разрабатываете SMTP-приложение, обязательно изучите конструкцию всех кодов SMTP-ответа. То, как коды составлены в самом SMTP – превосходный образец грамотного подхода к делу. В табл.2 приведены возможные значения кодов ответа SMTP, определенные в RFC 821.

Таблица 2. Коды ответа SMTP и их значение

Код	Значение
211	Ответ о состоянии системы или помощь
214	Сообщение-подсказка (помощь)
220	<имя_домена> служба готова к работе
221	<имя_домена> служба закрывает канал связи
250	Запрошенное действие почтовой транзакции успешно завершилось
251	Данный адресат не является местным; сообщение будет передано по маршруту <forward-path>
354	Начинай передачу сообщения. Сообщение заканчивается комбинацией CRLF-точка-CRLF
421	<имя_домена> служба недоступна; соединение закрывается
450	Запрошенная команда почтовой транзакции не выполнена, так как почтовый ящик недоступен
451	Запрошенная команда не выполнена; произошла локальная ошибка при обработке сообщения
452	Запрошенная команда не выполнена; системе не хватило ресурсов
500	Синтаксическая ошибка в тексте команды; команда не опознана
501	Синтаксическая ошибка в аргументах или параметрах команды
502	Данная команда не реализована
503	Неверная последовательность команд
504	У данной команды не может быть аргументов
550	Запрошенная команда не выполнена, так как почтовый ящик недоступен
551	Данный адресат не является местным; попробуйте передать сообщение по маршруту <forward-path>
552	Запрошенная команда почтовой транзакции прервана; дисковое пространство, доступное системе, переполнилось
553	Запрошенная команда не выполнена; указано недопустимое имя почтового ящика
554	Транзакция не выполнена

В настоящее время почта Internet заметно меняется. Почта Internet состоит из трех частей: конверт, заголовки и тело сообщения. Появляются новые SMTP команды, которые изменяют конверт, в заголовках могут быть использованы не-ASCII символы, а к телу добавляются структуры (MIME).

Изменения в конверте: расширенный SMTP.

RFC 1425 [Klensin et al. 1993a] определяет основные расширения для SMTP. В результате получилось то, что называется расширенным SMTP (ESMTP – extended SMTP). Как и другие новые характеристики (уже описанные в тексте), эти изменения добавляются таким образом, что новые версии совместимы со старыми.

Клиент, которому необходимо использовать новые характеристики, открывает сессию с сервером с использованием команды EHLO вместо команды HELO. Совместимый сервер отвечает откликом с кодом 250. Этот отклик обычно содержит несколько строк, причем каждая строка содержит ключевое слово и дополнительный аргумент. Ключевые слова указывают на расширения SMTP, поддерживаемые сервером. (В случае многострочного отклика во всех строках кроме последней, между цифровым кодом и текстом стоит дефис. В последней строке между цифровым кодом и текстом стоит пробел.) Вновь появляющиеся расширения будут описаны в RFC и будут зарегистрированы в IANA.

## 6. Промежуточные агенты.

Термин "маршрут доставки" (forward-path) служит для того, чтобы отличать почтовый ящик (mailbox), имя которого абсолютно, от пути (он может быть различным), по которому следует почта. Предположим, что мы хотим доставить два почтовых сообщения на один и тот же сетевой компьютер. Оба сообщения имеют один и тот же адрес, однако не обязательно будут следовать по одному и тому же маршруту. Точно так же, если на пришедшие сообщения выдаются ответы, они не обязательно будут следовать по указанному обратному маршруту (reverse-path). Как правило, конкретный маршрут для почты выбирается системным администратором. Чтобы направить почту по нужному пути, используются значения маршрута доставки и обратного маршрута, в которых указываются промежуточные агенты (relay agents). Промежуточный агент доставки – это MTA, так называемый почтовый хаб (mail hub), настроенный на передачу транзитной почты. Чтобы доставить сообщение, местный агент пользователя (UA) передает его местному MTA, который, в свою очередь, передает его промежуточному агенту MTA. В следующем примере Smith@usc.edu является почтовым ящиком, а HOST1, HOST2 и HOST3 – промежуточными агентами:

```
MAIL FROM:<@HOST1, @HOST2, @HOST3:Smith@usc.edu>
```

В наше время промежуточные агенты присутствуют практически во всех сетях, входящих в Internet. На рис.3 приведена типичная конфигурация почтовой системы Internet с участием промежуточных агентов.

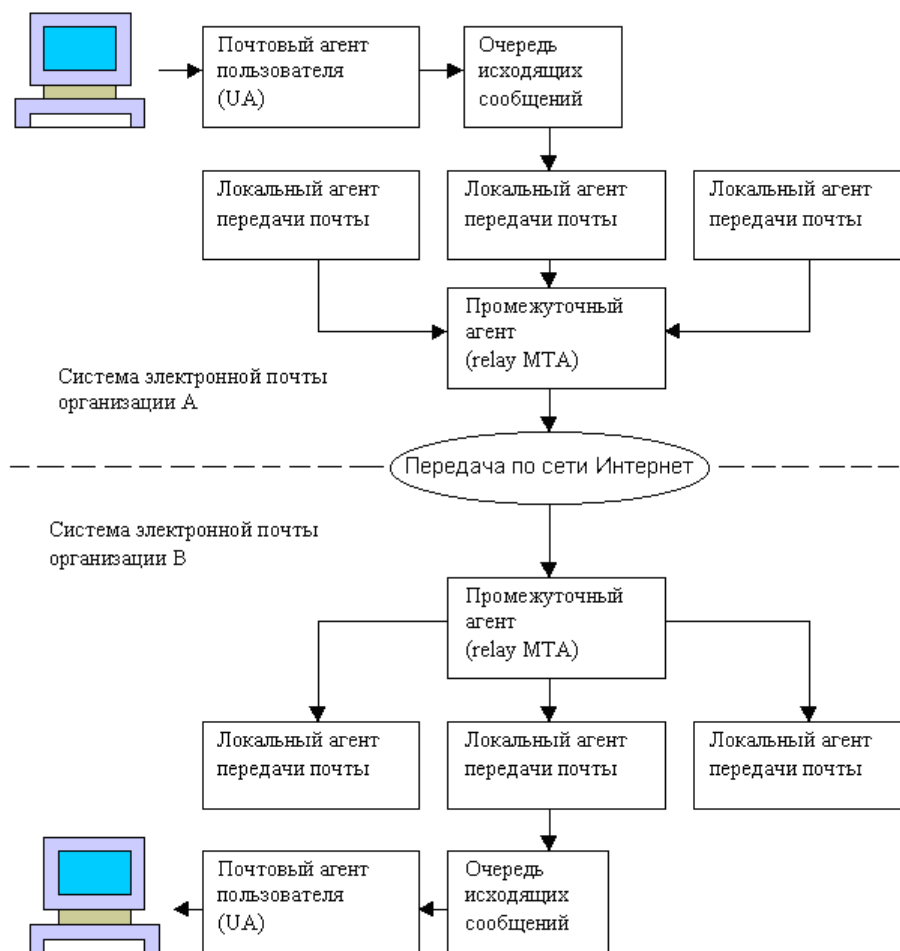


Рисунок 3. Почтовая система Internet с участием промежуточных агентов.

Чтобы упростить процесс конфигурации почтовой системы, в локальной сети устанавливается один компьютер, служащий промежуточным агентом (relay host). Вся почта пользователей попадает сначала на него. Затем этот компьютер рассылает сообщения по Internet. Кроме всего прочего, такой компьютер может служить защитой от злоумышленников. Ограничивая общение локальной сети с внешним миром до уровня почты, организация сводит до минимума риск нежелательного вторжения в свои собственные системы.

Кроме того, администрировать и защищать в этом случае приходится единственный компьютер. SMTP в состоянии послать сообщение непосредственно с компьютера пользователя на компьютер адресата в том случае, если между ними существует прямое почтовое соединение. К сожалению, это далеко не всегда так. Как правило, между двумя компьютерами находится один или несколько промежуточных агентов. Чтобы обеспечить доставку, в почтовом сообщении нужно указать имя компьютера-получателя и точное наименование почтового ящика. Аргументом команды MAIL является обратный маршрут, включающий имя источника сообщения и имена всех промежуточных агентов. Аргумент команды RCPT – маршрут доставки, содержащий имя получателя сообщения. Обратный маршрут описывает путь, который прошло сообщение, тогда как маршрут доставки идентифицирует место назначения. Обратный маршрут используется SMTP, когда нужно передать сообщение о случившейся ошибке или о невозможности доставить

сообщение, когда оно уже прошло через промежуточный агент. По мере продвижения сообщения по Internet записи о его маршрутах изменяются. В обязанности системных администраторов входит правильно настраивать местные МТА на передачу сообщений промежуточному агенту, и наоборот, промежуточные агенты на доставку сообщений местным МТА. Если у промежуточного МТА изменится имя, все, что нужно сделать в конфигурации местного МТА – изменить имя компьютера в системе DNS. Другие параметры конфигурации не изменяются. Другими словами, повторим еще раз, что иметь один компьютер для промежуточной доставки – значит, снять с себя значительную часть головной боли по настройке почтовой системы – ведь придется заботиться только об одном компьютере.

Рассмотрим почтовую транзакцию между промежуточными агентами SMTP. До того как сообщение будет передано следующему указанному в маршруте (в поле TO:) компьютеру, имя данного компьютера удаляется из маршрута доставки и добавляется в начало обратного маршрута. К тому моменту, когда сообщение достигнет пункта назначения, маршрут доставки будет содержать только имя почтового ящика. В RFC 821 приведен пример того, как изменяется содержимое маршрутов по мере обработки почтового сообщения. Когда промежуточный агент А получает почту со следующими аргументами:

```
FROM: USERX@HOSTY.ARPA
```

```
TO: <@HOSTA.ARPA, @HOSTB.ARPA: USERC@HOSTD.ARPA>
```

он переправляет почту сетевому компьютеру В со следующими аргументами:

```
FROM: <@HOSTA.ARPA: USERX@HOSTY.ARPA> TO: <@HOSTB.ARPA: USERC@HOSTD.ARPA> .
```

Как видим, промежуточный агент А (HOSTA.ARPA) убрал свое имя из заголовка <TO:> и добавил в заголовок <FROM:>. Промежуточный агент компьютера В совершит аналогичное действие, и следующим пунктом назначения сообщения будет почтовый ящик USERC на компьютере HOSTD.ARPA.

**Примечание:** Другими словами, обратные маршруты и маршруты доставки строятся агентами передачи почты по мере прохождения сообщения от одного агента к следующему. Если очередной на пути сообщения SMTP-агент не умеет обслуживать промежуточную доставку, он должен ответить таким же кодом, какой предусмотрен на случай отсутствия местного почтового ящика.

## 7. Формат почтового сообщения.

При обсуждении примеров отправки и получения почтовых сообщений уже упоминался формат почтового сообщения. Разберем его подробнее. Формат почтового сообщения Internet определен в документе RFC-822 (Standard for ARPA Internet Text Message). Это довольно большой документ объемом в 47 страниц машинописного текста, поэтому рассмотрим формат сообщения на примерах. Почтовое сообщение состоит из трех частей: конверта, заголовка и тела сообщения. Пользователь видит только заголовок и тело сообщения. Конверт используется только программами доставки. Заголовок всегда находится перед телом сообщения и отделен от него пустой строкой. RFC-822 регламентирует содержание заголовка сообщения. Заголовок состоит из полей. Поля состоят из имени поля и содержания поля. Имя поля отделено от содержания символом ":". Минимально необходимыми являются поля Date, From, cc или To, например:

```
Date: 26 Aug 76 1429 EDT
```

```
From: Jones@Registry.org
```

```
cc:
```

или

```
Date: 26 Aug 76 1429 EDT
```

```
From: Jones@Registry.org
```

```
To: Smith@Registry.org
```

Поле Date определяет дату отправки сообщения, поле From – отправителя, а поля cc и To – получателя(ей). Чаще заголовок содержит дополнительные поля:

```
Date: 26 Aug 76 1429 EDT
```

```
From: George Jones<Jones@Registry.org>
```

```
Sender: Secy@SHOST
```

```
To: Smith@Registry.org
```

```
Message-ID: <4231.629.XYzi-What@Registry.org>
```

В данном случае поле Sender указывает, что George Jones не является автором сообщения. Он только переслал сообщение, которое получил из Secy@SHOST. Поле Message-ID содержит уникальный идентификатор сообщения и используется программами доставки почты. Следующее сообщение демонстрирует все возможные поля заголовка:

```
Date: 27 Aug 76 0932
```

```
From: Ken Davis <Kdavis@This-Host.This.net>
```

```
Subject: Re: The Syntax in the RFC
```

```
Sender: KSecy@Other-host
```

```
Reply-To: Sam.Irving@Reg.Organization
```

```
To: George Jones <Jones@Registry.org>
```

```
cc: Important folks:
```

```
Tom Softwood <Balsa@Tree.Root>,
```

```
"Sam Irving"@Other-Host;
```

```
Standard Distribution:
```

```
/main/davis/people/standard@Other-Host
```

```
Comment: Sam is away on business.
```

```
In-Reply-To: <some.string@DBM.Group>, George's message
```



X-Special-action: This is a sample of user-defined field-names.

Message-ID: <4331.629.XYzi-What@Other-Host

Поле Subject определяет тему сообщения, Reply-To – пользователя, которому отвечают, Comment – комментарий, In-Reply-To – показывает, что сообщение относится к типу "В ответ на Ваше сообщение, отвечающее на сообщение, отвечающее ...", X-Special-action – поле, определенное пользователем, которое не определено в стандарте.

Следует сказать, что формат сообщения постоянно дополняется и совершенствуется. В RFC-1327 введены дополнительные поля для совместимости с почтой X.400. Кроме этого, следует обратить внимание на поля некоторых довольно часто встречающихся заголовков, которые не регламентированы в RFC-822. Так первое предложение заголовка, которое начинается со слова From, содержит UUCP-путь сообщения, по которому можно определить, через какие машины сообщение "пробиралось". Поле Received: содержит транзитные адреса почтовых серверов с датой и временем прохождения сообщения. Вся эта информация полезна при разборе трудностей с доставкой почты.

## Задание на работу

1. Настроить почтовый сервер на базе exim (postfix) для приема почтовых сообщений для домена согласно варианту задания. Обеспечить возможность отправки почтовых сообщений только для пользователей, прошедших аутентификацию, а также с заданных в варианте узлов. Включить поддержку "черных списков" и псевдонимов почтовых ящиков.
2. Включить возможность отправки и приема почтовых сообщений поверх SSL/TLS.
3. Проанализировать с помощью утилиты telnet/openssl протокольный обмен между клиентом/сервером и сервером SMTP.

### Варианты заданий.

№ варианта	Имя домена	Узлы, отправляющие почту без аутентификации	Почтовый ящик	
			Имя	Псевдоним
1	zone01.com.ua	172.20.1.10 172.20.1.20 172.20.1.30 172.20.1.40 172.20.1.50	alpha	ws1
			beta	ws2
			gamma	ws3
			delta	ws4
			omega	ws5
2	zone02.net.ua	192.168.11.21 192.168.11.22 192.168.11.23 192.168.11.24 192.168.11.25	mercury	srv01
			venus	srv02
			earth	srv03
			saturn	srv04
			jupiter	srv05
3	zone03.kiev.ua	192.168.1.11 192.168.1.12 192.168.1.13 192.168.1.14 192.168.1.15	tiger	www
			lion	ftp
			lynx	nnntp
			leopard	pop3
			jaguar	imap
4	zone04.com	172.20.1.31 172.20.2.32 172.20.3.33 172.20.4.34 172.20.5.35	rose	machine1
			tulip	machine2
			narcissus	machine3
			aster	machine4
			peony	machine5
5	zone05.net	172.25.11.10 172.25.11.20 172.25.11.30 172.25.11.40 172.25.11.50	alpha	srv01
			beta	srv02
			gamma	srv03
			delta	srv04
			omega	srv05
6	zone06.org.ua	192.168.22.10 192.168.22.20 192.168.22.30 192.168.22.40 192.168.22.50	mercury	www
			venus	ftp
			earth	nnntp
			saturn	pop3
			jupiter	imap

№ варианта	Имя домена	Узлы, отправляющие почту без аутентификации	Почтовый ящик	
			Имя	Псевдоним
7	zone07.org	172.30.10.31	tiger	machine1
		172.30.10.32	lion	machine2
		172.30.10.33	lynx	machine3
		172.30.10.34	leopard	machine4
		172.30.10.35	jaguar	machine5
8	zone08.edu	192.168.33.1	rose	ws1
		192.168.33.2	tulip	ws2
		192.168.33.3	narcissus	ws3
		192.168.33.4	aster	ws4
		192.168.33.5	peony	ws5
9	zone09.org	192.168.55.10	alpha	www
		192.168.55.20	beta	ftp
		192.168.55.30	gamma	nnntp
		192.168.55.40	delta	pop3
		192.168.55.50	omega	Imap
10	zone10.org.ua	172.21.11.10	mercury	machine1
		172.21.11.20	venus	machine2
		172.21.11.30	earth	machine3
		172.21.11.40	saturn	machine4
		172.21.11.50	jupiter	machine5
11	zone11.net	172.31.50.1	tiger	ws1
		172.31.50.2	lion	ws2
		172.31.50.3	lynx	ws3
		172.31.50.4	leopard	ws4
		172.31.50.5	jaguar	ws5
12	zone12.com	192.168.77.11	rose	srv01
		192.168.77.22	tulip	srv02
		192.168.77.33	narcissus	srv03
		192.168.77.44	aster	srv04
		192.168.77.55	peony	srv05
13	zone13.kiev.ua	192.168.99.10	alpha	machine1
		192.168.99.20	beta	machine2
		192.168.99.30	gamma	machine3
		192.168.99.40	delta	machine4
		192.168.99.50	omega	machine5
14	zone14.net.ua	172.16.70.1	mercury	ws1
		172.16.70.2	venus	ws2
		172.16.70.3	earth	ws3
		172.16.70.4	saturn	ws4
		172.16.70.5	jupiter	ws5
15	zone15.com.ua	192.168.88.11	tiger	srv01
		192.168.88.12	lion	srv02
		192.168.88.13	lynx	srv03
		192.168.88.14	leopard	srv04
		192.168.88.15	jaguar	srv05

### *Литература*

1. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 4-е изд. – СПб.: Питер, 2010. – 944 с.
2. Стивенс У.Р. Протоколы TCP/IP. Практическое руководство/ Пер. с англ. и коммент. А.Ю. Глебовского. – СПб.: «Невский диалект» – «БХВ–Петербург», 2003. – 672 с.: ил.