Министерство науки и высшего образования Российской Федерации
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

П.В. МИЩЕНКО

ОСОБЕННОСТИ ФУНКЦИОНИРОВАНИЯ ПРОТОКОЛОВ ЭЛЕКТРОННОЙ ПОЧТЫ

Утверждено Редакционно-издательским советом университета
в качестве учебного пособия

НОВОСИБИРСК
2018

УДК 004.773.3.057.4(075.8)
М 717

Рецензенты:
д-р техн. наук, профессор *М.Г. Гриф*
канд. техн. наук, доцент *А.А. Якименко*

Мищенко П.В.

М 717 Особенности функционирования протоколов электронной почты: учебное пособие / П.В. Мищенко. – Новосибирск: Изд-во НГТУ, 2018. – 71 с.

ISBN 978-5-7782-3730-8

Учебное пособие разработано на основе материалов лекций и лабораторных занятий по курсам «Прикладные протоколы Интернета» и «Информационные сети», проводимых на факультете Автоматики и вычислительной техники ФГБОУ ВО «НГТУ».

Пособие, прежде всего, предназначено для подготовки специалистов и бакалавров по направлениям «Информатика и вычислительная техника» и «Программная инженерия». Кроме того, изложенный материал может быть успешно использован в процессе подготовки к олимпиадам и конкурсам в области сетевых информационных технологий.

УДК 004.773.3.057.4(075.8)

ISBN 978-5-7782-3730-8

ВВЕДЕНИЕ

Интернет – последнее и важнейшее из достижений XX века. Он изменил мир и отразился на жизни каждого человека, как бы далек тот ни был от компьютеров и компьютерных сетей. По своему воздействию на общество Интернет сопоставим с телефонной связью, телевидением, радио и печатной прессой вместе взятыми. Впрочем, это и не удивительно, ведь он уже впитал в себя все достижения человечества как в информационной сфере, так и в коммуникационной отрасли.

Среди базовых пользовательских технологий особое место занимает электронная почта, которая позволяет в считанные минуты переслать сообщение, содержащее как текстовые, так и звуковые, и графические, и программные файлы.

Электронная почта (E-mail) – одна из первых служб Интернета. Решение сетевой задачи передачи данных начинается с работы протокола прикладного уровня. Далее данные последовательно проходят по всему стеку и достигают физического уровня, который управляет их непосредственной передачей по физическому каналу связи. Абонент, принимающий данные, обрабатывает их аналогичным образом: передавая вверх по стеку до прикладного уровня.

Общее количество прикладных протоколов велико и продолжает расти. Ориентиром протоколов прикладного уровня служат конкретные прикладные задачи. Они определяют как процедуры по организации взаимодействия определенного типа между прикладными процессами, так и форму представления информации в рамках взаимодействия. В процессе обучения технологиям вычислительных сетей вызывает затруднения практическая часть исследования телекоммуникационных систем: построение топологии сети, настройка интерфейсов

оборудования, анализ взаимодействия сетевых протоколов. Причинами этому являются высокая стоимость оборудования, размещение сетевых устройств и организация рабочих мест обучающихся. Принимая во внимание факты, приведенные выше, возросла тенденция к появлению программного обеспечения, которое позволяет проводить моделирование процессов настройки и функционирования телекоммуникационных систем.

Благодаря симуляторам компьютерных сетей эксперименты в этой области можно проводить гораздо удобнее и экономнее, чем на реальном оборудовании. На базе кафедры Вычислительной техники Новосибирского государственного технического университета существует Лаборатория сетевых средств и технологий, оснащенная оборудованием «CISCO». В лаборатории студенты начинают работу с виртуальным оборудованием, которое реализовано в программном обеспечении Cisco. Cisco Packet Tracer – симулятор сети передачи данных, который предоставляет возможность студентам проводить эксперименты с сетью и оценивать вероятные сценарии. Являясь неотъемлемой частью комплексной среды обучения Сетевой академии, Packet Tracer предоставляет функции моделирования, визуализации, авторской разработки, а также облегчает процесс изучения сложных технологических основ.

Учебное пособие посвящено анализу процесса функционирования протоколов электронной почты. Пособие имеет практическую направленность и помимо теоретических сведений содержит примеры настройки прикладных протоколов в среде Packet Tracer. Учащимся предлагается применить и закрепить полученные знания на практике. Для этого в пособии представлены указания к выполнению практикума на тему «Анализ функционирования протоколов SMTP и POP3».

ПРОТОКОЛЫ ЭЛЕКТРОННОЙ ПОЧТЫ

В самом общем случае схема обмена электронными сообщениями (электронная почта, e-mail) выглядит следующим образом (рис. 1).

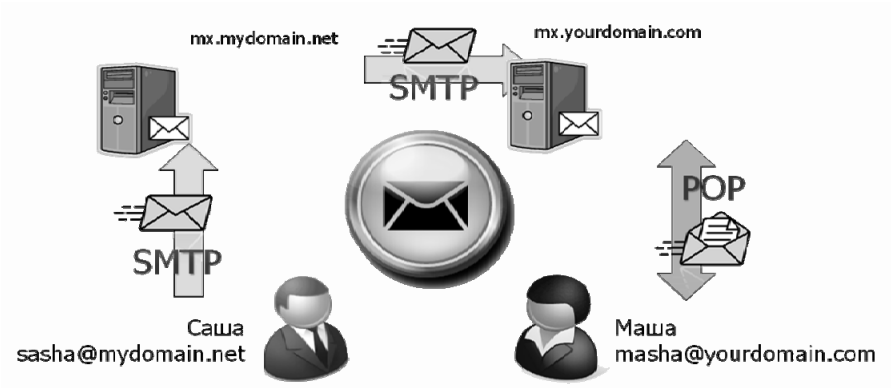


Рис. 1. Пример схемы обмена электронными сообщениями

В процессе обмена используются не только технические и программные средства отправителя и получателя, но и почтовые серверы, задачей которых является прием писем, предназначенных к отправке от отправителя, доставка их в почтовый ящик получателя, временное хранение, и передача из почтового ящика получателю. В качестве одного из элементов электронной почты, почтовый ящик представляет собой обычный каталог файловой системы (папку), а электронные письма – файлы данных, находящиеся в данном каталоге. Вся технология приема и передачи электронных писем подчиняется определенным правилам, задаваемым протоколами и форматами данных. На стороне

клиента (отправителя и получателя) используется специальное программное обеспечение – почтовый клиент, в качестве которого может использоваться, например, Microsoft Outlook для Windows или Mozilla Thunderbird для Linux. Даже если вы работаете со своим почтовым ящиком через веб-интерфейс (подключаясь к сайту, например mail.ru), то вы все равно используете почтовое клиентское программное обеспечение, выполняющееся в среде сервера. Почтовые серверы и почтовые клиенты, независимо от того, на каком оборудовании, и с каким программным обеспечением, они работают, реализуют, как минимум два прикладных протокола, без которых невозможен обмен почтой. Один из них служит для передачи электронных писем – это протокол SMTP (Simple Mail Transfer Protocol, простой протокол передачи почты), второй служит для приема POP3 (Post Office Protocol ver 3, протокол почтового офиса). Помимо этого, отдельно стоит отметить существование протокола IMAP (Internet Message Access Protocol).

Типы серверов

В момент отправления электронное письмо маршрутизируется с одного сервера на другой (такие серверы также называют почтовыми узлами, или релеями), и в результате доходит до почтового сервера получателя. Точнее говоря, сообщение отправляется на почтовый сервер, в задачу которого входит транспортировка сообщений:

MTA (англ. mail transfer agent) до MTA получателя. В Интернете MTA связываются друг с другом при помощи протокола SMTP, поэтому, логично, что они называются SMTP-серверами (или иногда серверами исходящей почты).

Затем MTA получателя доставляет электронное письмо на сервер входящей почты (называющийся MDA (англ. mail delivery agent), то есть агент доставки электронной почты, который хранит письмо в ожидании его приема пользователем (рис. 2).

Существует два основных протокола извлечения почты из MDA:

- POP3, более старый из двух, использующийся, чтобы извлечь письмо и, в определенных случаях, оставить его копию на сервере;
- IMAP, использующийся для координирования статуса сообщений (прочитано, удалено, перемещено) между многочисленными почтовыми клиентами. При использовании IMAP копия каждого письма

сохраняется на сервере, чтобы эта задача по синхронизации могла быть выполнена.

Серверы входящей почты получили названия POP-серверы и IMAP-серверы, в зависимости от используемого протокола.

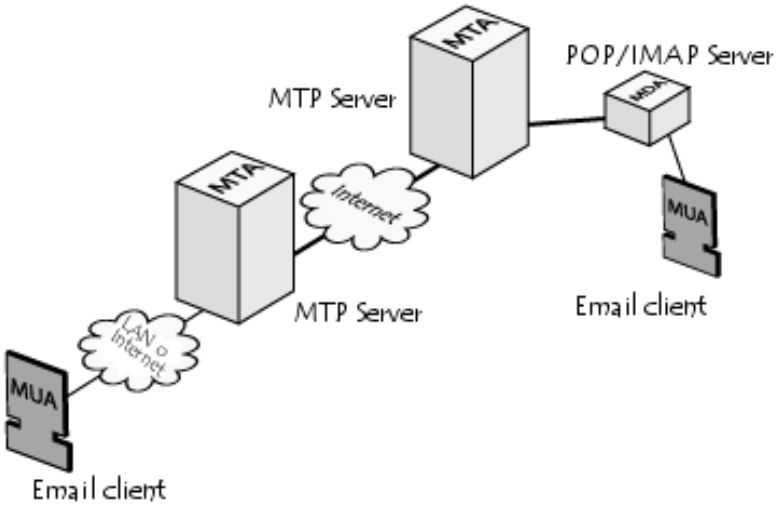


Рис. 2. Взаимодействие серверов

Используя аналогию с обычной почтой, можно сказать, что MTA выступают в качестве «почтовых отделений» (сортировка почты и доставка почтальоном), в то время как MDA выполняют роль почтовых ящиков, которые хранят сообщения (насколько позволяет их объем), пока получатель не проверит свой ящик.

Чтобы предотвратить проверку почты посторонними лицами, MDA защищен именем пользователя (логин) и паролем.

Извлечение почты осуществляется при помощи программы, которая называется MUA(англ. Mail User Agent). Если MUA установлен на компьютере пользователя,– это клиент электронной почты (такой как Mozilla Thunderbird, Microsoft Outlook, Eudora Mail, Incredimail или Lotus Notes). Если же в качестве MUA выступает веб-интерфейс, использующийся для взаимодействия с сервером входящей почты, он называется веб-почтой.

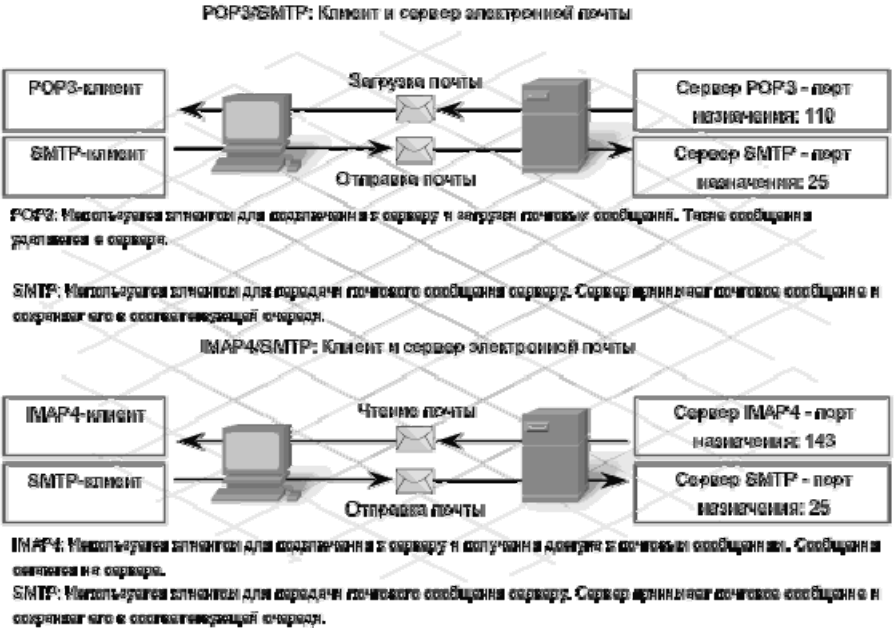


Рис. 3. Пример процесса отправки и доставки сообщения

На рис. 3 иллюстрируется сопоставление процессов отправки и доставки с использованием POP3/SMTP и IMAP4/ SMTP.

ПРОТОКОЛ SMTP

Simple Mail Transfer Protocol – это сетевой протокол, предназначенный для передачи электронной почты в сетях TCP/IP от клиента к серверу/ между серверами.

В качестве транспортного протокола SMTP использует TCP, соединение устанавливается через порт с номером 25.

Впервые протокол был описан в RFC 821 (1982 год), последнее обновление в RFC 5321 (2008) включает масштабируемое расширение **ESMTP** (англ. *Extended SMTP*). В настоящее время под «протоколом SMTP», как правило, подразумевают и его расширения.

На рис. 4 представлены этапы передачи сообщения.

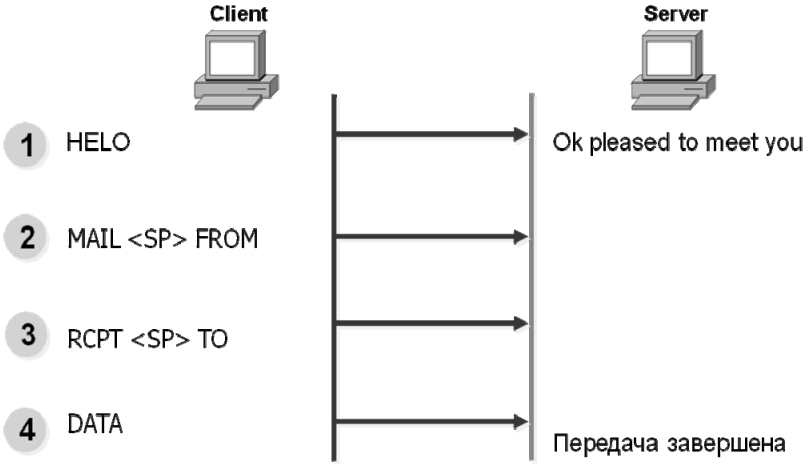


Рис. 4. Этапы передачи сообщения по SMTP

На рис. 5 представлен пример SMTP-сессии.

```
S: (ожидает соединения)
C: (Подключается к порту 25 сервера)
S: 220 mail.stankin.ru SMTP sendmail 8.13.5 is glad to see you!
C: HELO
S: 250 domain name should be qualified
C: MAIL FROM: <somebody@somecompany.com>
S: 250 somebody@somecompany.com sender accepted
C: RCPT TO:<user1@stankin.ru>
S: 250 user1@stankin.ru ok
C: RCPT TO: <user2@stankin.ru >
S: 550 user2@stankin.ru unknown user account
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Hi!
C: .
S: 250 769947 message accepted for delivery
C: QUIT
S: 221 mail.stankin.ru SMTP sendmail 8.13.5 closing connection
S: (закрывает соединение)
```

Рис. 5. Пример SMTP-сессии (S – сервер, C – клиент)

Основные команды SMTP

HELO – идентифицирует SMTP-сервер отправителя, открывает сеанс.

```
C: HELO user.example.net
S: 250 server.example.com Hello user.example.net
[192.168.1.1] pleased to meet you
```

Рис. 6. Команда HELO

MAIL FROM – задает адрес отправителя.

```
C: MAIL FROM: <user@example.net>
S: 250 2.1.0 user@example.net... Sender ok
```

Рис. 7. Команда MAIL FROM

RCPT TO – задает адрес получателя.

```
C: RCPT TO: <user2@example.com>
S: 250 2.1.5 user2@example.com... Recipient ok
```

Рис. 8. Команда RCPT TO

DATA – указывает на начало сообщения. Для окончания сообщения указывается точка.

```
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: This is a test message.
C: .
S: 250 2.0.0 I3PDY91f000484 Message accepted for
delivery
```

Рис. 9. Команда DATA

QUIT – завершает SMTP-сеанс.

```
C: QUIT
S: 221 2.0.0 server.example.com closing connection
```

Рис. 10. Команда QUIT

ESMTP

Протокол SMTP изменился незначительно. На смену команде HELO, использовавшейся для начала диалога, пришла команда EHLO, позволяющая работать с расширениями ESMTP. Команды, применяемые для настройки почтовых систем и для получения справочной информации о пользователях, теперь используются значительно осторожнее, чем в восьмидесятые годы. Эти команды, к сожалению, создают удобства не только для сетевых администраторов, но и для злоумышленников. Поэтому такие команды обычно используют только на этапе настройки почтовой системы. В работающей системе их, как правило, отключают.

Главной целью протокола SMTP является надежная и эффективная доставка электронных почтовых сообщений. Для реализации протокола требуется только надежный канал связи. Средой для SMTP может служить отдельная локальная сеть, система сетей или же всемирная сеть Internet.

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, который выполняет некоторые дополнительные функции. Промежуточный сервер в таких случаях действует как шлюз в другие среды передачи и выбирается обычно с использованием MX-записей DNS (служба доменных имен).

Протокол SMTP базируется на следующей модели коммуникаций: в ответ на запрос пользователя почтовая программа-отправитель сообщения устанавливает двустороннюю связь с программой-приемником (почтовым сервером). Получателем может быть оконечный или промежуточный адресат. Если необходимо, почтовый сервер может установить соединение с другим сервером и передать сообщение дальше.

Для того чтобы получить сообщение из своего почтового ящика, почтовая программа пользователя соединяется с сервером уже не по протоколу SMTP, а по специальному почтовому протоколу получения

сообщений. Такой протокол позволяет работать с почтовым ящиком: забирать сообщения, удалять сообщения, сортировать их и выполнять другие операции. Самым популярным в настоящее время протоколом такого рода является протокол Post Office Protocol v.3 (POP3).

Многие концепции, принципы и понятия протокола POP3 выглядят и функционируют подобно SMTP: взаимодействие происходит посредством команд. Сервер POP3 находится между агентом пользователя и почтовыми ящиками.

Он предусматривает соединение с почтовым сервером на основе транспортного протокола TCP через порт 110. Спецификация POP3 определена в документе RFC 1939. POP3 разработан с учетом специфики доставки почты на персональные компьютеры и имеет соответствующие операции для этого.

Конструкция протокола POP3 обеспечивает возможность пользователю обратиться к своему почтовому серверу и изъять накопившуюся для него почту. Пользователь может получить доступ к POP3-серверу из любой точки доступа к Internet. При этом он должен запустить специальный почтовый агент, работающий по протоколу POP3, и настроить его для работы со своим почтовым сервером. Сообщения доставляются клиенту по протоколу POP3, а посылаются при помощи SMTP. То есть на компьютере пользователя существуют два отдельных агента-интерфейса к почтовой системе – доставки (POP3) и отправки (SMTP).

POP3

POP3 (англ. Post Office Protocol Version 3) – почтовый протокол, используемый почтовым клиентом для получения сообщений электронной почты с сервера.

POP (POP1) определён в RFC 918 (1984), POP2 в RFC 937 (1985). Первоначальная спецификация POP3 была представлена в RFC 1081 (1988).

Нынешняя же описана в RFC 1939, обновлена механизмом расширения (RFC 2449) и механизмом аутентификации (RFC 1734).

Версии POP2 был назначен порт 109.

Изначальная спецификация POP3 поддерживала только незашифрованный механизм входа в систему USER/PASS или управление доступом .rhosts. На данный момент, POP3 поддерживает различные методы аутентификации для предоставления разных уровней защиты от незаконного доступа к пользовательской почте. Большинство из них предоставлены механизмами расширения POP3. Было высказано неофициальное предложение для спецификации «POP4», с рабочей реализацией сервера. Это предложение добавило основные функции управления папками, поддержку составных сообщений, а также управление флагами сообщений. Однако, никакого прогресса «POP4» не наблюдается с 2003 г. Для транспорта POP3 использует 110 порт TCP.

POP поддерживает простые требования «загрузи-и-удали» для доступа к удалённым почтовым ящикам. Хотя большая часть POP-клиентов предоставляет возможность оставить почту на сервере после загрузки, использующие POP клиенты обычно соединяются, извлекают все письма, сохраняют их на пользовательском компьютере как новые сообщения, удаляют их с сервера, после чего разъединяются.

На рис. 11 представлены примеры POP3-сессии.

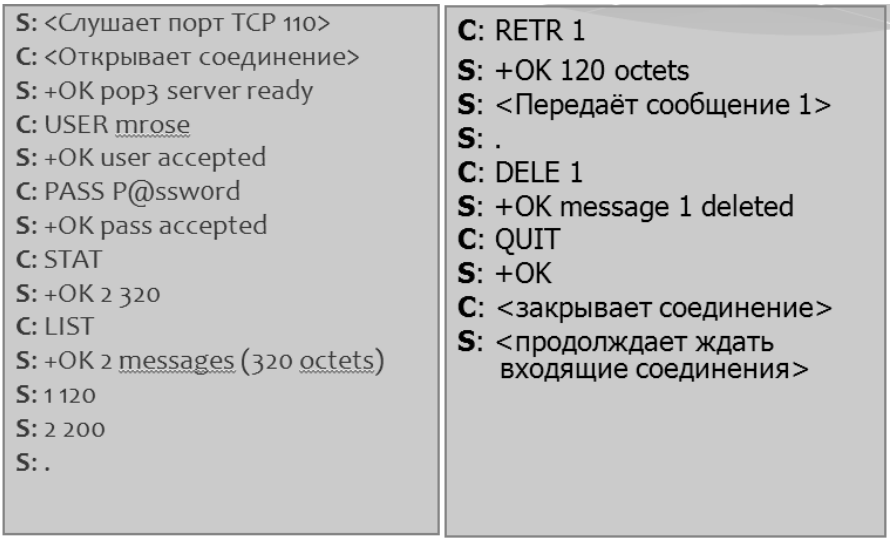


Рис. 11. Пример POP3-сессии

Основные команды POP3

USER [имя] – передаёт серверу имя пользователя.
Аргументы:
[имя] – строка, указывающая имя почтового ящика.
Ограничения - нет.
Возможные ответы:

- +OK name is a valid mailbox,
- -ERR never heard of mailbox name.

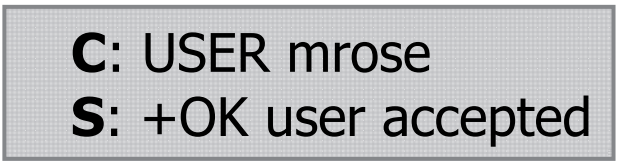


Рис. 12. Команда USER

PASS [пароль] – передаёт серверу пароль почтового ящика.
Аргументы:
[пароль] – пароль для почтового ящика.
Ограничения:
работает после успешной передачи имени почтового ящика.
Возможные ответы:

- +OK maildrop locked and ready,
- -ERR invalid password ,
- -ERR unable to lock maildrop.

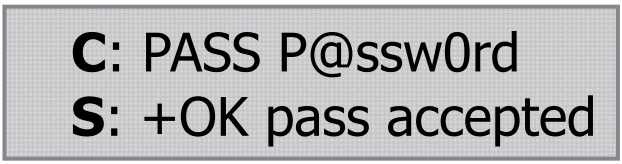


Рис. 13. Команда PASS

APOP [имя] [digest] – команда служит для передачи серверу имени пользователя и зашифрованного пароля (digest).

Аргументы:

[имя] – строка, указывающая имя почтового ящика.

[digest] – зашифрованная временная метка паролем пользователя по алгоритму MD5.

Ограничения:

Её поддержка не является обязательной.

Возможные ответы:

- +OK maildrop has n message,
- -ERR password supplied for [имя] is incorrect.

C: APOP mrose c4c9334bac560ecc979e58001b3e22fb
S: +OK mrose's maildrop has 2 messages (320 octets)

Рис. 14. Команда APOP

STAT – сервер возвращает количество сообщений в почтовом ящике плюс размер почтового ящика.

Аргументы **– нет**.

Ограничения – доступна после успешной идентификации.

Возможные ответы:

+OK a b

C: STAT
S: +OK 2 320

Рис. 15. Команда STAT

LIST [номер сообщения] – запрос информации о сообщении (или о всех – без аргумента).

Аргументы:

[сообщение] – номер сообщения (необязательный аргумент)

Ограничения – доступна после успешной идентификации.

Возможные ответы:

- +OK scan listing follows,
- -ERR no such message.

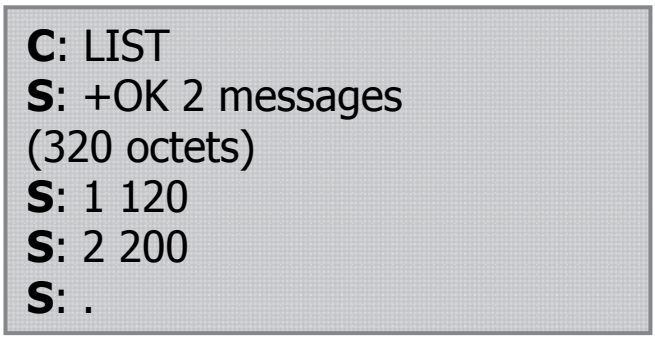


Рис. 16. Команда LIST

RETR [номер сообщения] – сервер передаёт сообщение с указанным номером.

Аргументы:

[сообщение] – номер сообщения.

Ограничения – доступна после успешной идентификации.

Возможные ответы:

- +OK message follows,
- -ERR no such message.

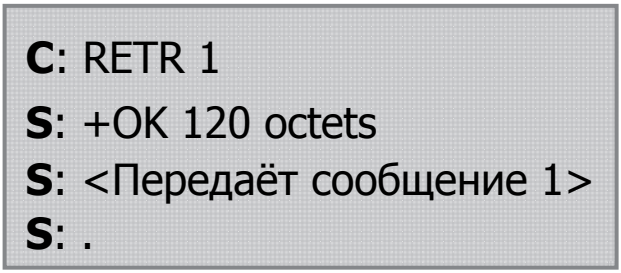


Рис. 17. Команда RETR

DELE [номер сообщения] - сервер помечает указанное сообщение для удаления.

Аргументы:

[сообщение] – номер сообщения.

Ограничения – доступна после успешной идентификации.

Возможные ответы:

- +OK message deleted,
- -ERR no such message.

Рис. 18. Команда DELE

QUIT

Аргументы – нет.

Ограничения – нет.

Возможные ответы:

+OK

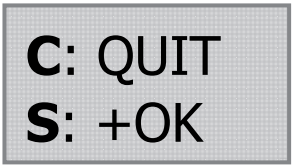


Рис. 19. Команда QUIT

IMAP

IMAP (англ. – Internet Message Access Protocol) позволяет клиентам получать доступ и манипулировать сообщениями электронной почты на сервере. Рассмотрим хронологию развития протокола и описывающие его спецификации. Появление первой версии датируется 1986 г., IMAP2 – 1988г. – RFC 1064, 1990 г. – RFC 1176, IMAP3 – 1991 г. – RFC

1203, IMAP2bis – 1993 г. IMAP4 (переименованный IMAP2bis), IMAP4rev1 (2001 г. – RFC 2822, 2003 г. – RFC 3501).

IMAP4 имеет 2 режима работы: online и offline. Особенности:

- Допускает одновременное подключение нескольких клиентов к 1 ящику;
- Возможность частичной загрузки сообщений;
- Возможность поиска на сервере.

Рис. 20 иллюстрирует принцип работы протокола.

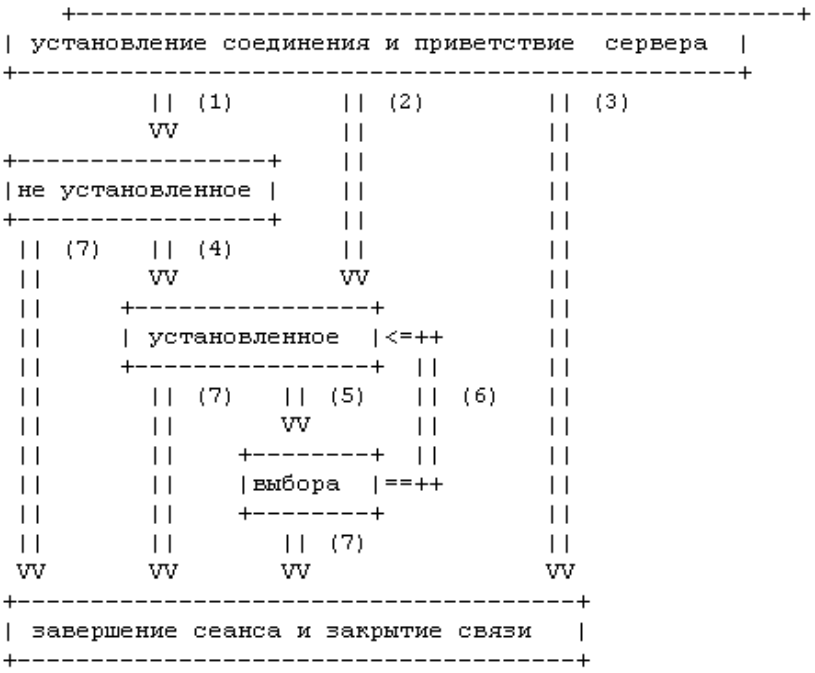


Рис. 20. Принцип работы протокола IMAP

На рис. 20 обозначены:

(1) связь без установления подлинности (приветствие OK)

(2) связь перед регистрацией (приветствие PREAUTH)

(3) отклоненная связь (приветствие BYE)

(4) успешное выполнение команды LOGIN или AUTHENTICATE

(5) успешное выполнение команды SELECT или EXAMINE

(6) команда CLOSE, или не успешное выполнение команды SELECT или EXAMINE

(7) команда LOGOUT, отключение сервера или прерывание связи

Любая процедура начинается с команды клиента. Любая команда клиента, в свою очередь, начинается с префикса-идентификатора (обычно короткая буквенно-цифровая строка, например A0001, A0002 и т. д.), называемого меткой (tag). Для каждой команды клиент генерирует свою метку. Сервер посылает запрос продолжения команды, если он готов. Такой отклик сервера начинается с символа "+".

Данные, передаваемые сервером клиенту, а также статусные отклики, которые не указывают на завершение выполнения команды, имеют префикс "*" и называются непомеченными откликами.

Выделяют три вида отклика завершения сервера:

* ok (указывает на успешное выполнение),
* no (отмечает неуспех),
* bad (указывает на протокольную ошибку, например, не узнана команда или зафиксирована синтаксическая ошибка).

Команды клиента

CAPABILITY – запрашивает перечень возможностей, поддерживаемых сервером.

Таблица 1

Параметры команды CAPABILITY

Аргументы:	отсутствуют	
Отклики:	необходим немаркированный отклик: CAPABILITY.	
Результат:	OK	успешное завершение команды;
	BAD	команда неизвестна или неверный аргумент.

NOOP

Таблица 2

Параметры команды NOOP

Аргументы:	отсутствуют.	
Отклики:	никакого специального отклика на эту команду не требуется.	
Результат:	OK	команда успешно завершена;
	BAD	команда неизвестна или неверен аргумент.

LOGOUT – информирует сервер о том, что клиент прерывает соединение. Сервер должен послать немаркированный отклик BYE, прежде чем отсылать маркированный отклик OK, после чего завершить разрыв соединения.

Таблица 3

Параметры команды LOGOUT

Аргументы:	отсутствуют.	
Отклики:	необходим немаркированный отклик BYE.	
Результат:	OK	прерывание сессии завершено;
	BAD	неизвестная команда или неверный аргумент.

Команды клиента в состоянии без аутентификации

В состоянии без аутентификации команды AUTHENTICATE или LOGIN организуют аутентификацию и переводят систему в состояние с аутентификацией. Об аутентификации в IMAP можно прочесть в документе RFC-1731. Команда AUTHENTICATE предоставляет общий

механизм для целого ряда методов аутентификации, среди которых команда LOGIN используется для традиционного ввода имени и пароля в текстовом виде.

AUTHENTICATE

Таблица 4

Параметры команды AUTHENTICATE

Аргументы:	имя механизма аутентификации (м.б. название метода шифрования).	
Отклики:	может быть запрошена дополнительная информация.	
Результат:	OK	Аутентификация завершена, осуществлен переход в состояние аутентификация выполнена;
	NO	Ошибка аутентификации: неподдерживаемый механизм аутентификации, параметры аутентификации отвергнуты;
	BAD	Неизвестная команда или неверный аргумент, механизм аутентификации прерван.

Команда AUTHENTICATE указывает серверу на механизм аутентификации, как это описано в [IMAP-AUTH]. Если сервер поддерживает запрошенный механизм аутентификации, он выполняет обмен согласно аутентификационному протоколу и идентифицирует клиента. Он может также согласовать опционный механизм защиты для последующих протоколов взаимодействия. Если запрошенный механизм аутентификации не поддерживается, сервер должен отвергнуть команду AUTHENTICATE путем посылки маркированного отклика NO.

LOGIN – идентифицирует клиента серверу и передает пароль пользователя открытым текстом.

Таблица 5

Параметры команды LOGIN

Аргументы:	имя пользователя, пароль.	
Отклики:	команда не требует какого-либо специального отклика.	
Результат:	OK	login завершено, система в состоянии с аутентификацией;
	NO	login не прошла: имя пользователя или пароль отвергнуты;
	BAD	команда неизвестна или неверный аргумент.

Команды клиента в состоянии «аутентификация осуществлена»

В состоянии «аутентификация осуществлена» разрешены команды манипуляции почтовыми ящиками как объектами-атомами. Команды SELECT и EXAMINE реализуют выбор почтового ящика и переход в состояние «выбрано».

SELECT

Таблица 6

Параметры команды SELECT

Аргументы:	имя почтового ящика.	
Отклики:	Необходимы немаркированные отклики: FLAGS, EXISTS, RECENT;	
	опционны немаркированные отклики OK: UNSEEN, PERMANENTFLAGS.	
Результат:	OK	процедура выбора закончена, система находится в состоянии выбрано;
	NO	выбор неудачен: нет такого ящика, доступ к почтовому ящику невозможен;
	BAD	команда неизвестна или неверен аргумент.

Команда SELECT осуществляет выбор почтового ящика, так, чтобы обеспечить доступ к сообщениям, находящимся там. Прежде чем присылать клиенту OK, сервер должен послать клиенту следующие немаркированные данные:

FLAGS – флаги, определенные для почтового ящика,

<n> EXISTS – число сообщений в почтовом ящике,

<n> RECENT – число сообщений с набором флагов \Recent,

OK [UIDVALIDITY <n>] – уникальный идентификатор корректности.

EXAMINE – идентична команде SELECT и дает тот же результат, однако, выбранный почтовый ящик идентифицируется «только для чтения». Никакие изменения постоянного состояния почтового ящика в этом случае не разрешены. Текст маркированного отклика OK на команду EXAMINE должен начинаться с кода отклика [READONLY].

Таблица 7

Параметры команды EXAMINE

Аргументы:	имя почтового ящика.	
Отклики:	Необходимы немаркированные отклики: FLAGS, EXISTS, RECENT;	
	опционны немаркированные отклики OK: UNSEEN, PERMANENTFLAGS.	
Результат:	OK	Просмотр закончен, система в состоянии выбор сделан;
	NO	Просмотр не прошел, система в состоянии аутентификация выполнена; нет такого почтового ящика; доступ к почтовому ящику невозможен;
	BAD	Команда неизвестна или неверен аргумент.

CREATE – создает почтовый ящик с заданным именем. Отклик OK присылается в случае, когда новый почтовый ящик с указанным

именем создан. Попытка создания INBOX или почтового ящика с именем существующего почтового ящика является ошибкой. Любая ошибка при попытке создания почтового ящика вызовет маркированный отклик NO.

Таблица 8

Параметры команды CREATE

Аргументы:	имя почтового ящика.	
Отклики:	На эту команду не посылается каких-либо специфических откликов.	
Результат:	OK	Команда выполнена;
	NO	команда не выполнена: почтовый ящик с таким именем не может быть создан;
	BAD	команда неизвестна или неверен аргумент.

DELETE – навечно удаляет почтовый ящик с указанным именем. При этом присылается маркированный отклик OK только в том случае, когда ящик уничтожен. Ошибкой считается попытка стереть INBOX или ящик с несуществующим именем.

Таблица 9

Параметры команды DELETE

Аргументы:	имя почтового ящика.	
Отклики:	Команда не требует каких-либо откликов.	
Результат:	OK	команда завершена;
	NO	ошибка при выполнении команды: не удается стереть ящик с этим именем;
	BAD	команда неизвестна или неверен аргумент.

UNSUBSCRIBE – удаляет специфицированный почтовый ящик из списка активных или подписных почтовых ящиков данного сервера, как это определяется командой LSUB. Эта команда возвращает маркированный отклик OK только в случае, если ликвидация подписки прошла успешно.

Таблица 10

Параметры команды UNSUBSCRIBE

Аргументы:	имя почтового ящика.	
Отклики:	Эта команда не требует каких-либо специфических откликов.	
Результат:	OK	ликвидация подписки прошла успешно;
	NO	ликвидация подписки не прошла: это невозможно для данного имени;
	BAD	команда неизвестна или неверен аргумент.

LIST – для получения списка подкаталогов, находящихся в каталоге и доступных клиенту, используется команда LIST. Аргументами команды являются: имя каталога, список необходимых подкаталогов (пустая строка – "" означает текущий каталог) и маска имен подкаталогов.

Команда LISTYLE="возвращает субнабор имен из полного набора, доступного клиенту. Присылается нуль или более немаркированных откликов LIST, содержащих атрибуты имен, иерархические разделители и имена.

Таблица 11

Параметры команды LIST

Аргументы:	имя,	
	имя почтового ящика может содержать символы подмены (wildcard).	
Отклики:	немаркированные отклики LIST.	
Результат:	OK	команда LISTYLE="выполнена;
	NO	команда не прошла: не возможно выполнение LISTYLE=" для данного образца или имени;
	BAD	команда неизвестна или неверен аргумент.

Служба DNS

Раздел посвящен изучению прикладных протоколов электронной почты SMTP и POP3. Однако взаимодействие с системой электронной почты невозможно без системы доменных имен (DNS). В задачи службы DNS входит:

1. Преобразование символических имен в IP-адреса;
2. Преобразование IP-адресов в символические имена.

Дополнительной функцией DNS является маршрутизация почты. Основная спецификация распределенной службы DNS указана в RFC 1034 и RFC 1035.

Единицами хранения и передачи информации в DNS являются ресурсные записи. Существует множество типов ресурсных записей, каждая из которых состоит из определенного числа полей. Для маршрутизации почты используется запись “MX”, при ее отсутствии запись типа “A”. Запись “A” (адресная запись) содержит параметры: доменное имя узла, соответствующий IP-адрес.

Пример: aivt IN A 195.19.212.16, где “IN” – это класс записи (Интернет).

Запись “MX” содержит параметры: имя почтового домена, имя почтового сервера, приоритет.

Пример: aivt IN MX 20 mail.stu.neva.ru, где “IN” – это класс записи (Интернет).

При получении письма MTA анализирует его служебную информацию, в частности заголовок письма, определяя домен получателя. Если он относится к домену, который обслуживается данным MTA, производится поиск получателя и письмо помещается в его ящик. Если домен получателя не обслуживается этим MTA, формируется DNS-запрос, запрашивающий MX-записи для данного домена. MX-запись представляет особый вид DNS-записи, которая содержит имена почтовых серверов, обрабатывающих входящую почту для данного домена. MX-записей может быть несколько, в этом случае MTA пробует последовательно установить соединение, начиная с сервера с наибольшим приоритетом. При отсутствии MX-записи запрашивается A-запись (запись адреса, сопоставляющая доменное имя с IP-адресом) и выполняется попытка доставить почту на указанный там хост. При невозможности отправить сообщение, оно возвращается отправителю (помещается в почтовый ящик пользователя) с сообщением об ошибке.

MIME

MIME (англ. – «Multipurpose Internet Mail Extensions») – в переводе на русский язык означает «Многоцелевые расширения почтового стандарта сети Интернет». Этот стандарт описывает, каким образом пересылать по электронной почте исполняемые, графические, мультимедийные и смешанные данные. Типичное применение MIME – пересылка графических изображений, аудио, документов Word, программ и даже просто текстовых файлов – то есть, когда важно, чтобы в ходе пересылки не производилось никаких преобразований над данными. MIME также позволяет разделять письмо на части различных типов так, чтобы получатель (почтовая программа) мог определить, что делать с каждой из частей письма. MIME – это фактически стандарт, но называется расширением, поскольку не противоречит прежнему стандарту RFC-822, а дополняет его.

Со времени опубликования в 1982 г., стандарт RFC 822 определил и полностью или частично внедрил формат текстовых писем в почтовой системе Internet. Но с расширением его использования, обнаружился ряд ограничений, заметно ограничивающих удовлетворение пользовательских потребностей. В частности, возможность пересылки нетекстовых данных, например, аудио и графики, просто не была упомянута в RFC822, описывавшем лишь формат текстовых сообщений. И даже в случае текста, RFC 822 обошел вниманием нужды пользователей, использующих расширенный набор символов, что характерно для азиатских и большинства европейских языков. Итак, требовалась дополнительная спецификация. Основное ограничение RFC822 - относительно короткие строки и 7-битная символьная таблица. Пользователям для отправки нетекстовых данных приходилось конвертировать тело своего письма в 7-битную форму с помощью UUENCODE, BINHEX и др.

Более очевидными стали ограничения RFC 822 при разработке почтовых шлюзов между хостами, использующими стандарт RFC822 и хостами, использующими стандарт X.400. X.400 имеет механизмы для включения нетекстовых данных в тело письма. В настоящее время стандарты для перевода почтовых сообщений из X.400 в RFC822 предполагают, что нетекстовые части тела письма должны быть сконвертированы (но не закодированы) в ASCII формат, либо должны быть "выброшены" из письма с уведомлением об этом получателя. А потеря информации крайне нежелательна для пользователя.

MIME разработан как расширяемый механизм с расчетом на то, что набор пар content-type/subtype будет расти со временем. Некоторые другие поля заголовка MIME, включая имена наборов символов, также, вероятно, получат большее число возможных значений. С этой целью MIME определяет процесс регистрации через Internet Assigned Numbers Authority (IANA), как центр регистрации этих значений.

1. Поле заголовка "MIME-Version"

Поскольку старый стандарт RFC 822 все еще используется, а MIME возможно, изменится и дополнится в будущем, почтовой программе необходимо знать, применен ли новый стандарт в конкретном письме или нет. Поэтому в заголовок введено новое поле "MIME-Version", объявляющее версию стандарта, в соответствии с которым написано данное письмо.

Все почтовые сообщения, составленные в соответствии с MIME-стандартом, должны иметь это поле в своем заголовке, например: MIME-Version: 1.0

Так как возможно, в будущем формат заголовка письма может расшириться, формально содержание поля "MIME-version" дается следующим образом:

версия := "MIME-Version" ":" 1*DIGIT "." 1*DIGIT

2. Поле заголовка "Content-Type"

Назначение этого поля - наиболее полное описание данных, содержащихся в теле, с тем, чтобы почтовый агент (программа) получателя могла выбрать соответствующий механизм для их обработки. Впервые это поле было определено в RFC 1049, но имело более простой синтаксис.

Данное поле включает в себя идентификаторы типа и подтипа, а также может содержать некоторую вспомогательную информацию, которая может потребоваться для конкретного типа данных. После идентификаторов типа и подтипа оставшаяся часть поля – просто набор параметров, заданных в порядке "атрибут/значение". Набор параметров зависит от типа данных. Очередность параметров значения не имеет. В числе определенных параметров – "charset", декларирующий символьный набор (кодировку, кодовую страницу – это все синонимы) тела письма. Комментарии допускаются. Вообще, поле Content-Type самого верхнего уровня используется для объявления общего типа данных, в то время как подтип определяет специальный формат для данных этого типа. Так, значение "image/xyz" поля Content-Type сообщает пользовательской программе, что данные являются графическим изображением (image), даже если эта почтовая программа не имеет понятия о специальном формате "xyz" этой картинки.

Пока имен типов только семь, и пока этого достаточно. Кроме того, предполагается, что расширение существующего набора поддерживаемых типов данных будет производиться за счет введения новых подтипов этих изначально определенных типов данных. В будущем добавление имен типов верхнего уровня может быть произведено только при принятии новой версии стандарта MIME. Если по какой-либо другой причине в существующей версии используется незарегистрированный тип содержимого, ему должно быть дано имя, начинающееся с

"X-", чтобы подчеркнуть его нестандартный статус и заранее предупредить конфликт с официальным именем типа, которое может быть введено позднее.

Правильное заполнение поля Content-Type:

содержимое := "Content-Type" ":" тип "/" подтип *(";" параметр)

; задание типа и подтипа (нечувствительное к регистру букв)

тип := "application" / "audio"

/ "image" / "message"

/ "multipart" / "text"

/ "video" / признак нестандартного типа

; Все значения нечувствительны к регистру букв

признак нестандартного типа := x- / iana-

iana- := <общепринятый признак расширения, зарегистрированный соответственно приложению "E" RFC 1521>

x- := <Два последовательных символа "X-" или "x-", без пробела

или другого символа между ними>

подтип := слово ; регистр безразличен

параметр := атрибут "=" значение

атрибут := слово ; регистр безразличен

значение := слово / строка в кавычках

слово := любые ASCII-символы кроме пробелов, Ctrl-последовательностей и специальных символов

Специальные символы:= "(" / ")" / "<" / ">" / "@"

/ "," / ";" / ":" / "" / <">

/ "/" / "[" / "]" / "?" / "="

; Эти символы используются в строке значений параметров

Здесь набор специальных символов отличается от набора, определенного в RFC 822 только наличием символов "/", "?", "=" и отсутствием символа ".".

Указание подтипа в данном поле является обязательным, т.к. нет подтипов по умолчанию. В отличие от имен типов, подтипов и параметров, значения параметров в общем случае являются чувствительными к регистру букв, но могут быть и нечувствительными - в зависимости от параметра. Например, значения границ multipart-письма являются чувствительными, а значение "access-type" для message/External-body не является.

Семь предопределенных типов верхнего уровня, более детально, представляют собой следующее:

text – текстовая информация. Основой подтип – **"plain"** – соответствует обычному неформатированному тексту и не требует специального программного обеспечения для отображения этого текста за исключением поддержки национальных кодировок. Другие подтипы используются в случае размеченного текста, когда с помощью специальной программы можно улучшить его визуализацию, но для понимания идеи содержания можно обойтись и без дополнительного ПО. Возможные подтипы могут описывать легко читаемые форматы различных текстовых процессоров.

multipart – содержимое письма состоит из некоторого множества частей, содержащих данные различных взаимно-независимых типов. Изначально определено четыре подтипа:

1. **"mixed"** – основной;
2. **"alternative"** – для представления одних и тех же данных в разных форматах;
3. **"parallel"** – если разные части документа должны просматриваться одновременно;
4. **"digest"** – если каждая из частей тела письма имеет тип "message".

message – письмо в письме. Тело, содержащее данные типа "message", само является письмом или частью письма, полностью отформатированного в соответствии со стандартом RFC 822, которое, в свою очередь, может содержать свое собственное поле заголовка "Content-Type".

Подтипы:

1. **"rfc822"** – основной;

2. **"partial"** – определен для частично-цитируемых писем для предотвращения фрагментирования тел содержащихся писем в случае слишком большой их общей длины для возможностей почтового транспорта;

3. **"External-body"** – используется, чтобы указать, что тело письма очень большое и находится вне письма.

image – графические данные. Графика требует соответствующего устройства вывода (графический дисплей, принтер, факс) для отображения своей информации. Изначально определены два подтипа для наиболее распространенных графических форматов – jpeg и gif.

audio – звуковая информация. Требует звуковое устройство (динамик или наушники) для вывода информации. Основной подтип – **"basic"**.

video – видео. Требует специальных аппаратных и программных возможностей для отображения видео-информации. Единственный изначально определенный подтип – **"mpeg"**.

application – как правило, не интерпретируемый двоичный код либо информация, предназначенная для обработки почтовой программой. Подтипы:

1. **"octet-stream"** – основной подтип; предназначен для не интерпретируемых двоичных данных, для которых рекомендуемым действием является предложение пользователю сохранить в файл на диске.

2. **"PostScript"** – дополнительный подтип; применяется при пересылке PostScript-документов в теле письма.

По умолчанию, письма, как и в стандарте RFC 822 пишутся простым (неразмеченным) текстом в языковой кодировке US-ASCII, что по спецификации MIME может быть описано как "Content-type: text/plain; charset=us-ascii". Это значение полагается, если поле Content-type не определено. Поэтому почтовая программа получателя может неверно истолковать содержимое письма, если при отправке не было указано поле Content-type, но на самом деле текст письма имеет другую кодировку или даже тип.

При отсутствии поля Content-type или поля MIME-Version в заголовке MIME-письма нельзя быть точно уверенным, что письмо имеет

языковую кодировку именно US-ASCII, поскольку могут еще встречаться почтовые программы, не использующие соглашения MIME. Но хотя возможно, что письмо, не содержащее в заголовке полей MIME-Version и Content-Type, может содержать все, что угодно, например, юниксовский tar-архив, сжатый gzip'ом и обработанный uuencode, все же, создателям почтовых программ рекомендуется оставлять этот факт без внимания и ориентироваться на значение по умолчанию, т.е. "text/plain; charset=us-ascii".[2]

Необходимо учесть, что в будущем ожидается заметное увеличение числа регистрированных типов и особенно подтипов содержимого писем. Если почтовая программа встретит неизвестное ей значение поля Content-type, она должна интерпретировать содержимое этого письма как "application/octet-stream" (см.выше).

3. Поле заголовка Content-Transfer-Encoding

Многие типы данных, пересылаемых через email требуют "натурального" представления, то есть, 8-битный набор символов либо двоичный код (что для машины - одно и то же, только представимо для пользователя по-разному). В таком виде данные не могут быть пересланы по 7-битным почтовым протоколам, например, RFC 821, который, к тому же, ограничивает длину строки 1000 символами. Стандартные механизмы конвертирования почты в 7-битный коротко строчный формат, приемлемый для почтового транспорта, описывает поле заголовка Content-Transfer-Encoding.

Значение поля должно быть строкой без пробелов, определяющей тип конвертации, как показано ниже:

конвертация := "Content-Transfer-Encoding" ":" механизм

механизм := "7bit" ;

/ "quoted-printable"

/ "base64"

/ "8bit"

/ "binary"

/ x-token

Значения не чувствительны к регистру букв. Значение "7BIT" означает, что тело письма уже имеет 7-битный формат и не требует дополнительной обработки для пересылки по почте. Это значение полагается по умолчанию, если поле заголовка Content-Transfer-Encoding отсутствует.

Значения "8bit", "7bit" и "binary" означают, что никакой трансформации содержимого не производится. Однако, они сделаны различными для индикации того, что из себя представляет содержимое письма, и, соответственно, способа обработки, который может потребоваться для данной транспортной системы. В частности:

"7bit" означает, что данные являются текстом, имеют короткие строки и языковую кодировку US-ASCII.

"8bit" означает короткие строки, но в них могут содержаться не-ASCII символы (128-255).

"Binary" означает, что тело письма может содержать не-ASCII символы, но строки могут быть произвольной длины, т.е. слишком длинными для SMTP-транспорта, и может не соблюдаться соглашение по признаку конца строки (CRLF), принятое в SMTP-транспорте.

Пять значений, определенных для поля Content-Transfer-Encoding, ничего не говорят о типе содержимого кроме указания алгоритма кодирования либо требований к почтовому транспорту в случае не кодированных данных.

Замечания по ограничениям конвертации

Необходимо предотвращать случаи вложенного кодирования, когда данные проходят через алгоритм конвертации несколько раз и должны столько же раз быть декодированы, чтобы быть читаемыми. Вложенное кодирование добавляет сложностей пользовательским почтовым программам: кроме очевидных проблем с множественной конвертацией, они могут скрыть основную структуру письма. В частности, они могут привести к тому, что несколько операций по декодированию могут потребоваться только для того, чтобы определить, объекты каких типов находятся в письме. Запрещение вложенного кодирования может осложнить работу некоторых почтовых шлюзов, но это будет меньшей проблемой по сравнению с трудностями для пользовательских почтовых программ.

3.1. Механизм конвертации "Quoted-Printable"

Этот механизм предназначен для представления данных, в основном состоящих из байтов, соответствующих символам, имеющим изображение в символьном наборе ASCII. В результате данного кодирования все байты будут иметь такие значения, гарантированные от дальнейшей модификации почтовым транспортом.

3.2. Механизм конвертации Base64

Этот алгоритм разработан для представления произвольных последовательностей байтов в форму, читаемую для человека. Кодирующий и декодирующий алгоритмы очень просты, но закодированные данные примерно на 33% больше, чем не кодированные. Base64 использует 65-символьный поднабор из US-ASCII, выделяя 6 бит на каждый печатный символ (65-й символ "=" используется для обозначения функции спец. обработки).

Процесс кодирования преобразует 4 входных символа в виде 24-битной группы, обрабатывая их слева направо. Эти группы затем рассматриваются как 4 соединенные 6-битные группы, каждая из которых транслируется в одиночную цифру алфавита base64. При кодировании base64, входной поток байтов должен быть упорядочен старшими битами вперед.

Каждая 6-битная группа используется как индекс для массива 64-х печатных символов. Символ, на который указывает значение индекса, помещается в выходную строку. Эти символы выбраны так, чтобы быть универсально представимыми и исключают символы, имеющие специальное значение для SMTP-транспорта (".", CR, LF) и для синтаксиса вложенных тел MIME ("-").

4. Дополнительные Content- поля заголовка

4.1. Необязательное поле Content-ID

При создании почтового агента верхнего уровня может быть желательно позволить одному телу иметь ссылку на другое. Для этого поля могут быть помечены с помощью поля заголовка "Content-ID", синтаксически идентичного полю "Message-ID":

идентификатор := "Content-ID" ":" идентификатор письма

Как и значения поля Message-ID, значения поля Content-ID должны быть абсолютно уникальными (для всего мира).

Такой идентификатор может быть использован для идентификации тела письма (части письма) в нескольких контекстах, в часности, для кэширования данных, указываемых с помощью механизма message/external-body. Хотя поле Content-ID является необязательным в общем случае, его использование необходимо в реализациях, генерирующих данные, имеющие дополнительный тип "message/external-body" (поле Content-type).

4.2. Необязательное поле Content-Description

Возможность ассоциировать некоторую описательную информацию с данными часто очень желательна. Например, может быть полезным описать тело, содержащее графическое изображение, как "a picture of the Space Shuttle Endeavor." Этот текст и может быть помещен в поле заголовка Content-Description.

описание := "Content-Description" ":" *текст

Описание должно иметь языковую кодировку US-ASCII, хотя механизм, определенный в RFC-1522 может быть использован для не-US-ASCII значений.

5. Предопределенные значения поля Content-Type

5.1. Тип 'Text'

Тип 'text' предназначен для пересылки текстовых материалов. Это значение поля - по умолчанию. Для обозначения языковой кодировки текста используется параметр "charset" для некоторых подтипов, включая основной подтип, "text/plain", соответствующий простому (неформатированному) тексту. В Internet'овской почте значением Content-Type по умолчанию является следующее: "text/plain; charset=us-ascii". Если текст является размеченным и нет соответствующего ПО для корректного визуального представления этого текста пользователю, имеет смысл сообщить ему подтип этих текстовых данных.

5.2. Тип 'multipart'

Этот тип используется, если один или более различных наборов данных заключены в одном письме. Каждая часть тела должна иметь синтакс письма RFC 822 (то есть, иметь заголовок, пустую строку и тело), но должна иметь открывающую и закрывающую границы.

5.3. Тип 'message'

При пересылке почты часто возникает необходимость включить внутрь письма другое письмо. Для этого и используется тип 'message'.

Основной подтип – "rfc822" – не требует параметров в поле Content-Type. Дополнительные подтипы – "partial" и "External-body" – предполагают наличие параметров.

5.4. Тип 'Application'

Этот тип используется для данных, не попадающих под остальные категории, в частности, для данных, обрабатываемых прикладными почтовыми программами. Это информация, которая должна быть обработана соответствующим приложением для того, чтобы принять наглядную либо исполняемую для получателя форму. Предполагаемое использование для этого типа включает в себя пересылку файлов по почте, таблицы, данные для почтовых систем расписания, языки для "активной" (вычислительной) почты.

5.5. Тип 'Image'

Этот тип означает, что тело письма содержит графический объект. Его подтипы соответствуют конкретным графическим форматам. Их значения нечувствительны к регистру букв. Два предопределенных подтипа – "jpeg" для формата JPEG с кодированием JFIF, и "gif" – для формата GIF.

Формальный синтаксис поля 'Content-Type':

image_тип := "image" "/" ("gif" / "jpeg" / подтип-расширение).

5.6. Тип 'Audio'

Этот подтип означает, что тело содержит аудио-данные. Хотя пока еще нет консенсуса по "идеальному" аудио-формату для компьютеров, сейчас имеется сильная потребность в универсальном формате.

Предопределенный подтип – "basic" введен в ответ на это требование, обеспечивая минимальный примитивный аудио-формат. Ожидается определение в дальнейшем более "богатых" форматов, обеспечивающих более высокое качество воспроизведения.

Содержимое тела, имеющего подтип "audio/basic", – аудио-данные в 8-битной форме, кодированные с использованием ISDN mu-law.

Формат, соответствующий этому подтипу, предполагает максимальную частоту звучания 8000 Hz и единственный канал.

Формальный синтаксис для поля 'Content-Type':

audio_тип := "audio" "/" ("basic" / подтип-расширение).

5.7. Тип 'Video'

Этот тип означает, что в теле письма содержится анимационное изображение, возможно, со звуком и цветом. Термин "video" используется безотносительно к технологии получения подвижного во времени изображения. Подтип "mpeg" указывает на видео, кодированное в соответствии со стандартом MPEG.

Хотя MIME-стандарт запрещает смешение разнородных мультимедийных данных в одном теле (письма или части письма), многие так называемые "video"-форматы включают синхронизированный звук, что допускается для подтипов типа "video".

Формальный синтаксис для поля 'Content-Type':

video-type := "video" "/" ("mpeg" / подтип-расширение).

5.8. Экспериментальные значения поля 'Content-Type'

Значение типа, начинающееся с "X-", считается частным, предназначенным для использования по договоренности между двумя или более почтовыми системами. Публично определенные (регистрированные) значения никогда не должны начинаться с префикса "X-".

В общем, использование X-типов верхнего уровня не рекомендуется. Производители почтового ПО должны по возможности обходиться использованием X-подтипов для предопределенных типов. Во многих случаях использование экспериментального подтипа более приемлемо, нежели типа верхнего уровня.

Значения полей Content-Type и subtype, а также другие параметры заголовка являются чувствительными к регистру букв, если только не оговорены исключения для конкретного значения параметра.

SPAM

Определение “Спама”

Спам (spam) – сообщения, массово рассылаемые людям, не дававшим согласие на их получение. Иными словами, спамом считается нежелательная, массовая, анонимная рассылка, причем каждый из вышеперечисленных критериев важен.

Не запрошенная означает то, что пользователь никоим образом не запрашивал полученное письмо, не получил его в ответ на свое, не подписывался на регулярную рассылку.

Массовая означает, что рассылка, которую мы считаем спамом, адресована множеству пользователей (тысячам, десяткам и сотням тысяч). Не запрошенную рассылку в адрес 10 пользователей мы не рассматриваем как спам.

Анонимная – значит, что для проведения рассылки использованы спамерские методы, скрывающие реальный адрес отправителя. Таким образом, рассылка, проведенная вручную со своего адреса (за исключением массовых рассылок) не будет отнесена к спаму.

Теперь интересно рассмотреть, что пользователи зачастую считают спамом, но что на деле таковым не является.

Рассылки, на которые пользователь подписался, даже если он забыл об этом – вполне легальный инструмент медиа или маркетинга, спамом не является. Отличительной чертой таких рассылок является возможность «отписаться» обязательно описанная в теле сообщения.

Автоматические ответы серверов (письмо доставлено, не доставлено, нет связи и т.д.) могут показаться назойливыми и ненужными, но это не спам. Для того, чтобы отфильтровать такие сообщения, в бизнес-продуктах предусмотрена отдельная категория – Formal.

Если вы пообщались на выставке с особо активным менеджером по продажам, после чего он осыпал вас коммерческими предложениями и follow-up’ами – это может казаться назойливым, но это не спам, так как не соблюдается критерий массовости и анонимности рассылки.

Ну и конечно же, сообщение от коллеги о продаже породистых котят, присланное на список рассылки по всей компании – письмо не запрошенное, массовое, но также не является спамом, будучи открытым

и не анонимным. И письмо от начальника о приближении дедлайна по проекту (при всей его нежелательности) – тоже не спам.

Основными видами спама являются:

- Реклама,
- Антиреклама,
- «Нигерийские письма»,
- Фишинг.

Истинное предназначение “Спама”

Для чего же рассылается настоящий спам? Кто в нем заинтересован? Ответ банален – для того, чтобы тем или иным способом заработать деньги.

Наиболее распространенная цель – реклама товаров и услуг. Учитывая, что во многих странах спам запрещен или появляются предпосылки к его законодательному запрету, – все большую долю в этой категории спама занимает реклама не обычных, а незаконных товаров – наркотики, поддельные лекарства, контрафактное ПО и т.д. В данном случае спамер выступает как продавец рекламного носителя, он не просто организует рассылку, он рекламирует деятельность заказчика (не свою, кроме случаев, когда рекламируется собственно «реклама в рассылках»).

Второй по величине тип спама – доля которого постоянно растет – это мошеннические рассылки. Наиболее известные их типы – это фишинг и «нигерийские письма». Фишинг – от англ. «ловить на удочку» - заключается в подделке письма таким образом, чтобы пользователь воспринял его как сообщение от банка (как правило), от своего ISP или системного администратора. Далее, если «наживка» сработала, и пользователь пребывает в уверенности, что общается с доверенной организацией – в письме ему предлагается под тем или иным предлогом сообщить данные своей банковской карты, счета, логины, пароли и т.д. Получив эти данные спамеры (фишеры) могут неплохо заработать.

«Нигерийские письма» – другой распространенный тип махинаций, когда пользователю пишет некий миллионер из маленькой страны, в которой произошла революция, и из которой срочно нужно вывести деньги. Чтобы сделать это, «миллионеру» непременно нужно воспользоваться банковским счетом пользователя. Естественно, что пользова-

тель получит «отличную комиссию». Далее несложно предположить, что происходит со счетом доверчивого получателя. С усилением антиспам-законодательства, спам все больше смещается из легальной рекламной – в нелегальную, в частности мошенническую сферу.

Другой нелегальный вид спама – вирусные рассылки. Для рассылки спама, спамерам постоянно нужны новые ПК, зараженные троянскими программами, которые и рассылают спам. Чтобы заполучить эти новые «зомби-ПК» и производятся рассылки с приложенными троянцами. Возможно и разделение труда: кто-то заинтересован в заражении ПК и заказывает спамеру рассылку вредоносного кода за деньги.

Спам также используют для PR-акций, распространения неконтролируемых слухов, создания ажиотажа – например, при игре на бирже. Некоторые политические силы также не брезгуют использовать спам для массовой пропаганды своих идей.

Технологии спама

В настоящее время выделяют следующие способы спам-рассылки:

- Прямая рассылка;
- Рассылка с выделенного сервера;
- Рассылка через dial-up провайдеров;
- Использование открытых релеев (Open Relay);
- Использование зомби-сетей;

Рассмотрим каждую из них подробнее.

Прямая рассылка

В большинстве случаев прямая массовая рассылка является первым шагом небольших компаний, решивших обратиться к спаму.

Для рассылки используется, как правило, тот же сервер (физический, а чаще виртуальный – на площадке провайдера), что и для обработки собственной почты.

Не используя каких-либо приемов для обмана спам-фильтров или сокрытия реального источника рассылки, такие «начинающие» спамеры легко идентифицируются провайдером и их действия пресекаются.

Рассылка с выделенного сервера

Этот вид рассылки спама – вариант предыдущего, но несколько более продвинутый.

Теперь уже устанавливается (или арендуется) мощный сервер для рассылки «в промышленных масштабах». Рассылка ведется с него.

Так же, как и в случае прямой рассылки с собственного рабочего сервера, эту рассылку можно легко идентифицировать хостинг-провайдеру. Однако некоторые из них смотрят сквозь пальцы на подобную практику, поскольку большую долю их клиентов составляют спамеры. Поэтому, подобный способ рассылки встречается и по сей день, хотя и в небольших объемах.

Рассылка через dial-up провайдеров

Каждый может пойти в магазин, купить скрэтч-карту dial-up доступа и анонимно получить доступ в Интернет. Легальный пользователь, который захочет использовать электронную почту воспользуется рекомендациями провайдера и станет отправлять почту через указанный ISP-сервер. Спамер же может настроить отправку почты напрямую получателю – минуя сервер провайдера. Это несложно, недорого и практически полностью анонимно. Такой метод широко использовался в 2000-2003 годах.

Сейчас же осуществить такой способ рассылки затруднительно. Провайдеры и общественные организации ведут списки IP адресов, зарезервированных под dial-up пользователей. Все они блокируются, вносятся в «черные списки», т.к. предполагается, что легальный пользователь будет посылать почту через сервер и IP, рекомендуемый провайдером.

Использование открытых релеев

Другой метод, популярный в 2000-2003 годах, но сейчас составляющий несколько процентов в общей доле рассылаемого спама – открытые релеи.

Открытый релей – это почтовый сервер, подключенный к сети Интернет, и позволяющий использовать себя для отправки писем без авторизации – т.е. любому пользователю извне. Такой сервер, как правило, является ошибочно настроенным по недосмотру или недостаточной квалификации его системного администратора.

Таким образом, спамер может использовать открытый релей для отправки с чужого оборудования и чужого IP (анонимность) массы своих сообщений.

В определенный момент открытые релеи стали настолько большой проблемой, что появились организации ведущие списки открытых релеев, информирующих их администраторов и вносящих их IP в черные списки. Этот процесс дал неплохие результаты, и сейчас количество почтовых релеев в Интернет очень невелико, как и процент спама рассылаемый таким способом.

Использование зомби-сетей

90 % спама и более на сегодняшний день рассылается с помощью так называемых зомби-сетей (бот-сетей, бот-нетов).

Зомби-сеть – это некоторое количество ПК в сети Интернет, никак не связанных друг с другом (они могут располагаться в разных сетях, странах и т.д.), при этом все эти ПК заражены одной и той же троянской программой, которая позволяет удаленно и незаметно управлять ими. В частности – рассылать спам, устраивать DDoS-атаки и т.д.

Рассылка спама с использованием зомби-сетей на порядок более сложная техническая задача, нежели описанные до этого методы. Поэтому к этому способу прибегают только профессиональные, квалифицированные, хорошо организованные команды спамеров. Поскольку этот способ требует создания и распространения вредоносных программ, наблюдается интеграция спамеров с вирусописателями – а также четкое «разделение труда». Кто-то создает троянскую программу, которая позволяет удаленно управлять, «зомбировать» ПК. Кто-то покупает эту программу, распространяет ее в спаме и/или через зараженные веб-сайты – и создает «зомби-сеть», размеры которой могут превышать 1 миллион ПК! Далее, «хозяин» «зомби-сети» может сам рассылать спам либо сдать сеть другим спамерам в аренду.

Процесс создания и работы «зомби-сети» может быть описан следующим образом:

1. Создается некая троянская программа, обладающая определенным функционалом по управлению зараженным ПК и сокрытием своего присутствия в системе.

2. Этот троян распространяется в Интернете и в спаме, и таким образом через небольшое время в Интернет появляется несколько тысяч (или сотен тысяч) зараженных ПК.

3. Зараженный ПК постоянно «слушает» определенный порт, ожидая команды от «управляющего центра». Как правило, для целей связи с Центром используется быстрый и анонимный протокол IRC, в своем нормальном применении используемый для веб-чатов.

4. В определенный момент сети поступает команда: взять контент с такой-то ссылки, установить незаметно от пользователя ПО для рассылки, разослать по такому-то списку адресатов.

5. После чего, собственно, «зомби-ПК» начинает незаметно от пользователя осуществлять спамерскую рассылку.

Спам несомненно является результатом неконтролируемой и непредсказуемо развивающейся глобальной сети. Существующие в настоящее время методы борьбы со спамом можно разделить по способу их организации на две категории: распределённые (контрольные суммы – сигнатуры; списки блокировки) и локальные (байесовская фильтрация; методы на основе формальных протокольных правил; процедурные методы; проверка подлинности отправителя). В отдельную категорию можно выделить закрытые методы, в том числе коммерческие. Среди широко известных средств борьбы со спамом стоит отметить: нераспространение e-mail адресов, фильтрацию сообщений, «Чёрные» списки, проверку имени-адреса, «Серые» списки».

ПРИМЕР НАСТРОЙКИ ПРОТОКОЛОВ SMTP И POP3

1. Построение топологии сети

Для исследования протоколов построим тестовую топологию сети следующего вида (рис. 21).

Произведем настройку сетевых устройств согласно заданным параметрам (табл. 12 и 13).

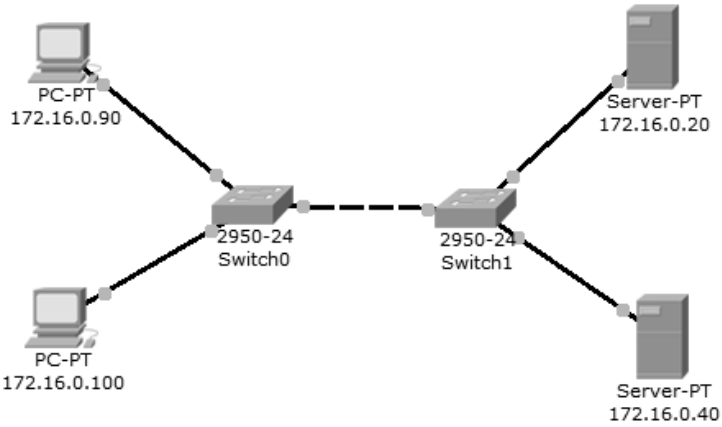


Рис. 21. Тестовая топология сети

Таблица 12

Параметры настойки ПК

Конечные узлы	IP-адрес	Маска сети	IP-адрес DNS-сервера
PC0	172.16.0.90	255.255.0.0	172.16.0.20
PC1	172.16.0.100	255.255.0.0	172.16.0.20

Таблица 13

Параметры настойки серверов

Серверы	IP-адрес	Маска сети	IP-адрес DNS-сервера
Server0	172.16.0.20	255.255.0.0	172.16.0.20
Server1	172.16.0.40	255.255.0.0	172.16.0.20

Все устройства расположены в одном сегменте локальной сети, поэтому маршрутизация пакетов не используется, значит, IP-адрес шлюза по умолчанию указывать необязательно.

2. Настройка почтового сервера

В качестве серверов электронной почты выступают сервер 172.16.0.20 и сервер 172.16.0.40. Схема взаимодействия с прикладными почтовыми протоколами применительно к построенной сети представлена на рис. 22.

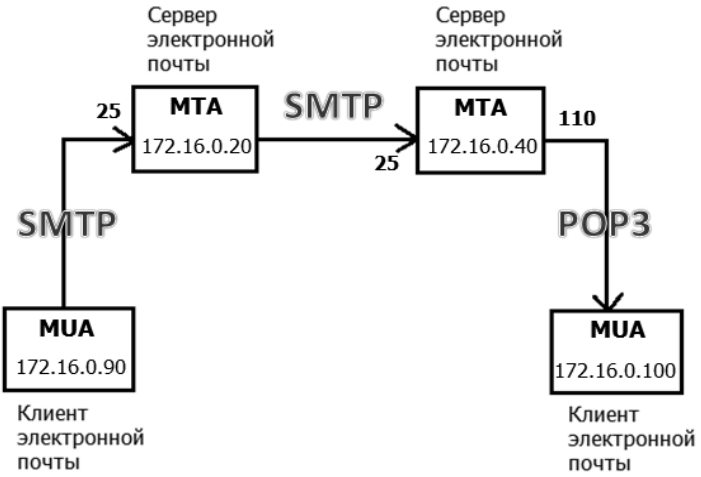


Рис. 22. Схема взаимодействия с прикладными почтовыми протоколами в исследуемой сети

На каждом из MTA будет поддерживаться smtp- и pop3-сервер. Подключиться к серверу может любой зарегистрированный пользователь. Чтобы отправить письмо, пользователь на сервере проходит авторизацию, после чего сервер готов отправлять письма от имени пользователя. По адресу назначения письма сервер определяет, кому следует передать его дальше. Нужный адрес сервер определяет с помощью службы DNS, в которой содержится соответствующая ресурсная адресная запись, преобразовывающая имя домена в IP-адрес.

Подключим службу DNS на сервере 172.16.0.20:

1) выполним один клик по выбранному устройству.

2) откроем вкладку *Config*, Services -> DNS (рисунок 23). Занесем данные о новой ресурсной записи: имя домена, IP-адрес, тип ресурсной записи. Симулятор не поддерживает ресурсную запись, предназначенную для почтовых серверов, MX, но ее можно заменить адресной (тип A).

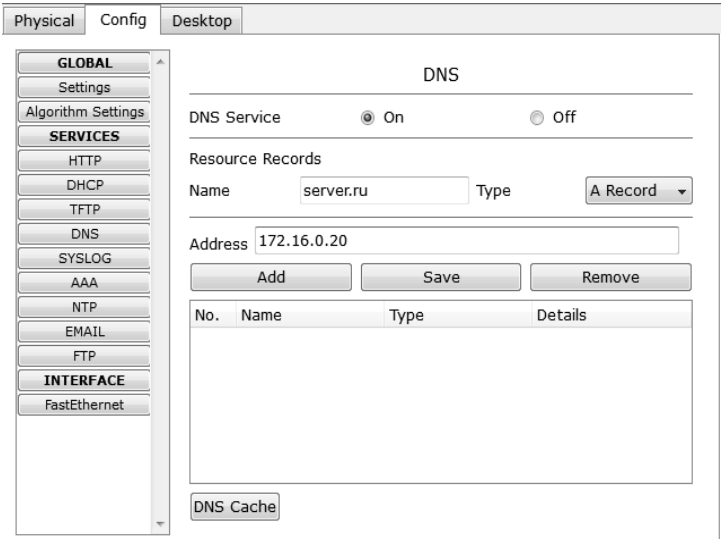


Рис. 23. Настройка службы DNS на сервере

3) нажимем на кнопку “Add” – будет добавлена новая запись в службу DNS (рис. 24).

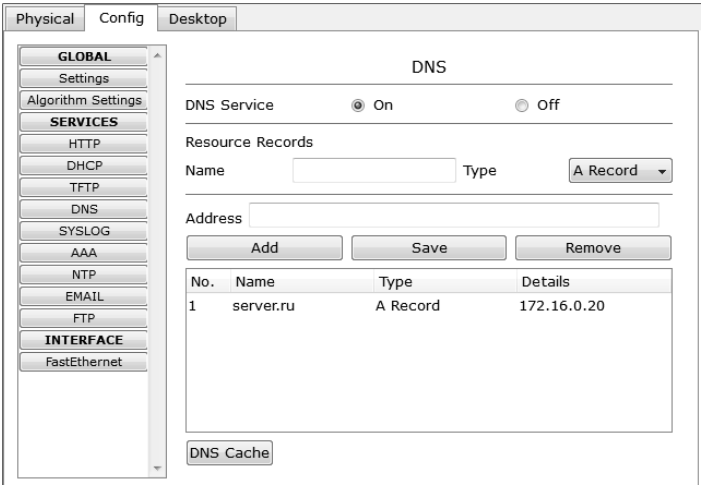


Рис. 24. Настройка службы DNS на сервере

Повторим предыдущие действия и добавим еще одну ресурсную запись о почтовом сервере 172.16.0.40 (рис. 25).

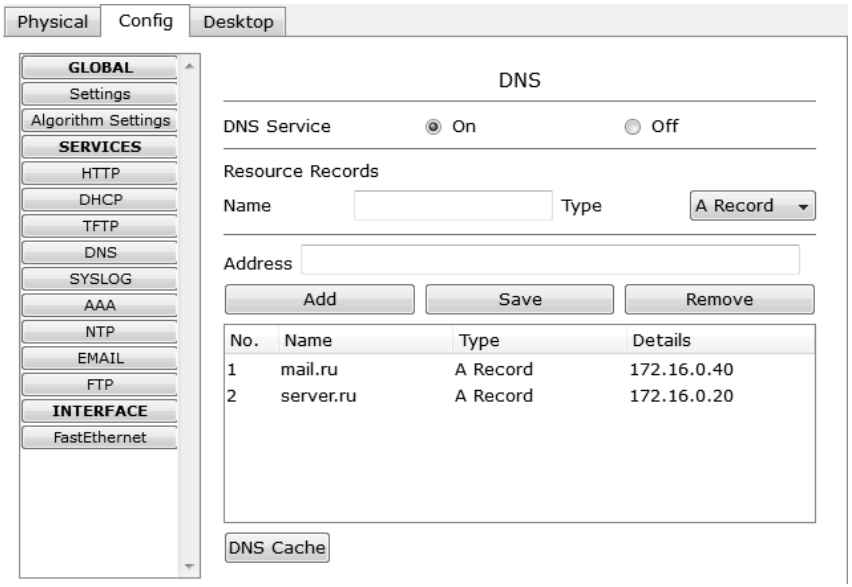


Рис. 25. Настройка службы DNS на сервере

Теперь сконфигурируем почтовый сервер 172.16.0.20 с поддержкой smtp- и pop3-сервера:

1) Выполним один клик по выбранному устройству.

2) Выбирем вкладку “Config”, Services -> EMAIL.

3) Подключим протоколы SMTP и POP3 и вводем имя домена электронной почты. Нажимем кнопку “Set” (рис. 26).

4) Создадим учетную запись для одного пользователя, вводем логин и пароль. Заносим запись в службу с помощью кнопки “+” (рис. 27).

SMTP-сервер и POP3-сервер на машине 172.16.0.20 сконфигурированы, имеют одного зарегистрированного пользователя. Так же на нем поддерживается служба DNS, в которой есть две ресурсных записи.

На сервере 172.16.0.40 тоже необходимо настроить почтовый сервер с поддержкой SMTP и POP3 (рисунок 28). В качестве DNS для него выступает сервер 172.16.0.20.

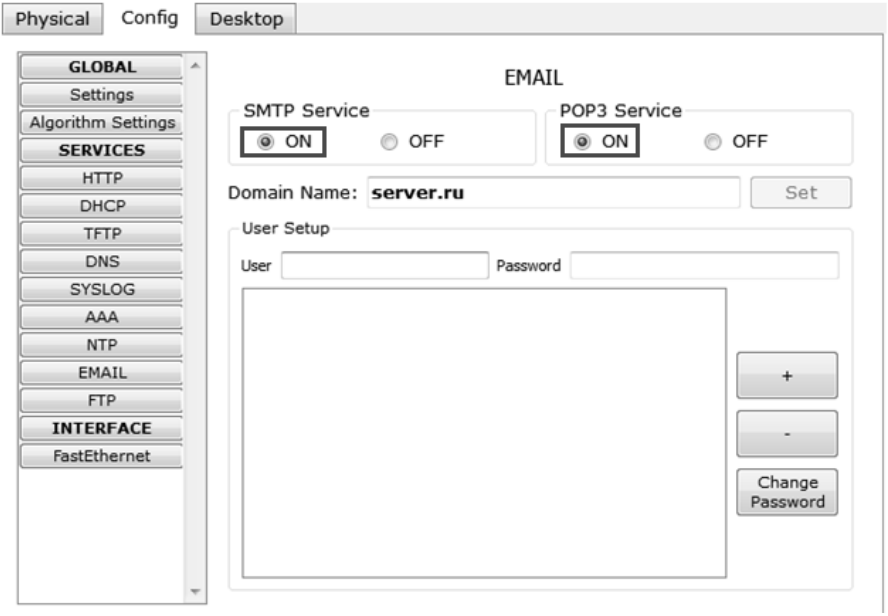


Рис. 26. Конфигурация smtp- и pop3-сервера

Рис. 27. Создание учетной записи

1) Один клик по выбранному устройству.
2) Выбираем вкладку “Config”, Services -> EMAIL

3) Подключаем протоколы SMTP и POP3 и вводим имя домена электронной почты - *mail.ru*. Нажимаем кнопку “Set”.

4) Создадим учетную запись для одного пользователя, вводим логин и пароль. Занести запись в службу можно с помощью кнопки “+”.

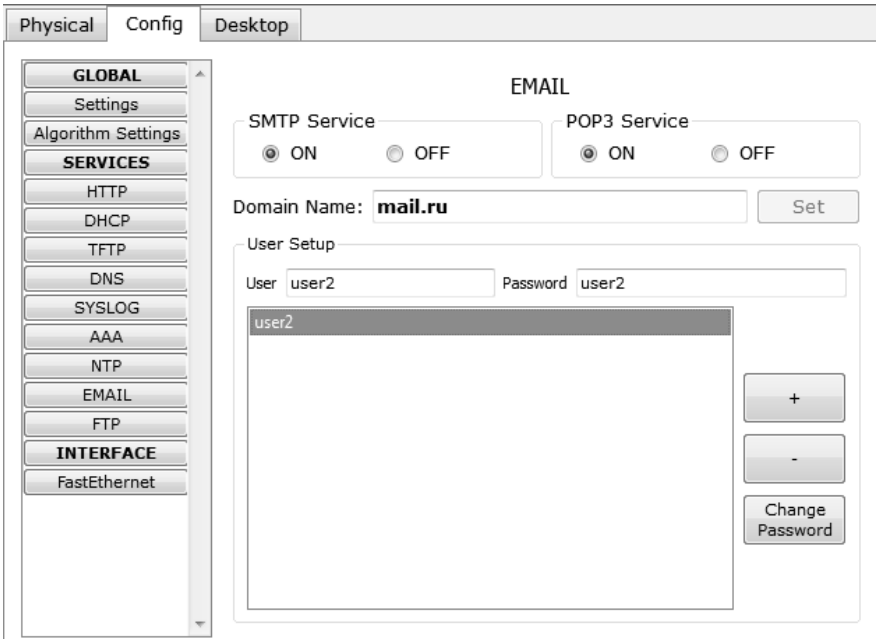


Рис. 28. Конфигурация smtp- и pop3-сервера

3. Настройка почтовой службы на конечных узлах

Для работы с почтовым smtp- или pop3-сервером на компьютере пользователя должен быть настроен клиент электронной почты, который и будет взаимодействовать с сервером.

Настроим на хосте 172.16.0.90 клиент электронной почты (рис. 29):

1) Один клик на хосте с IP-адресом 172.16.0.90.

2) Выбираем вкладку *Desktop*, программу “E-mail”. Появится окно конфигурации почтового сервиса. Вводим пользовательские данные в форму.

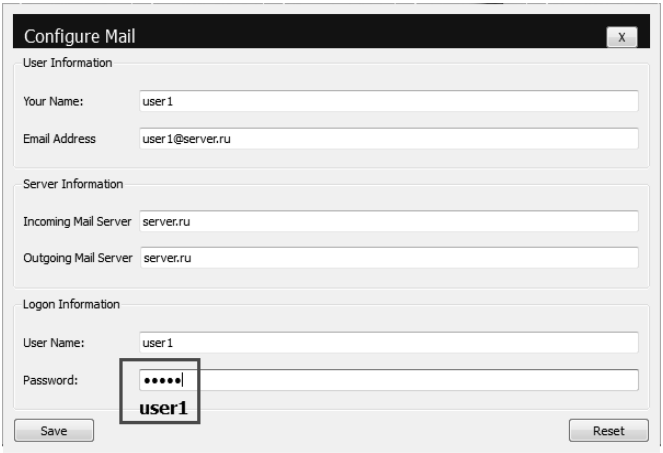


Рис. 29. Настройка клиента электронной почты

Нажимаем кнопку “Save”, закрываем окно, конфигурация клиента электронной почты завершена. Теперь для пользователя *user1* доступен почтовый сервис в домене *server.ru*: отправка и получение писем.

Настроим почтовый сервис и на хосте 172.16.0.100, выполнив предыдущие действия (рис. 30). Вводим следующие пользовательские данные:

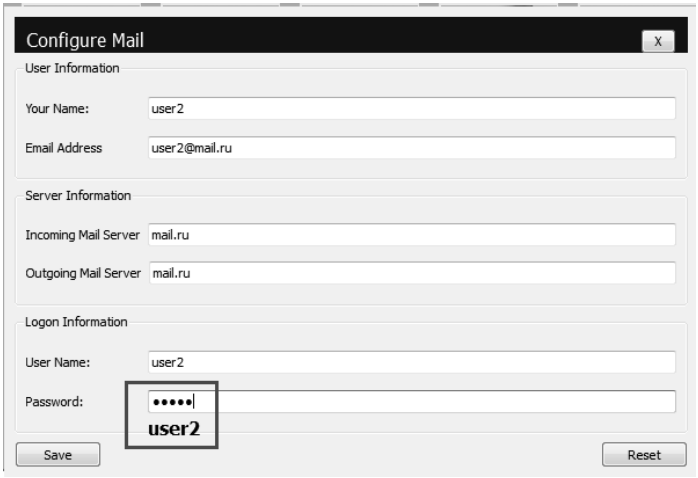


Рис. 30. Настройка клиента электронной почты

Теперь для пользователя *user2* доступен почтовый сервис в домене *mail.ru*: отправка и получение писем.

Настройка всех устройств и необходимых служб завершена.

4. Исследование прикладных почтовых протоколов в режиме симуляции

Переходим в режим симуляции Cisco Packet Tracer. Добавляем фильтры на 2 протокола: SMTP и POP3 (рис. 31). Это значит, что пакеты только фильтруемых протоколов будут отображаться в сети.

Рис. 31. Окно событий режима симуляции

Отправим письмо с хоста 172.16.0.90 от *user1* на хост 172.16.0.100 *user2* (рисунок 32):

1) Один клик по выбранному узлу (172.16.0.90).

2) Выбираем на вкладке “Desktop” программу “E-mail”.

3) Чтобы написать и отправить письмо, нажимаем на кнопку “Compose”. Появится форма, которую следует заполнить. В поле “To”

задается адрес электронной почты, кому вы отправляете письмо. Поле “Subject” содержит заголовок письма. Текст письма можете сочинить самостоятельно.

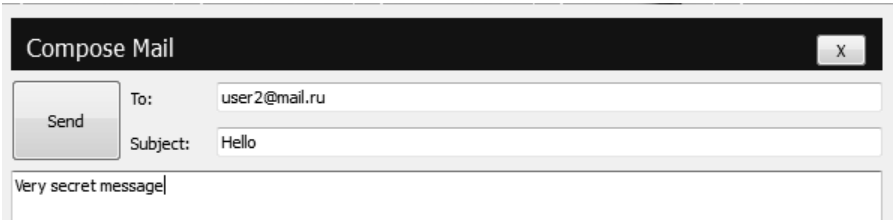


Рис. 32. Форма для отправления письма

Нажимаем на кнопку “Send”, начнется отправление письма.

Видим, что на хосте 172.16.0.90 сформировался пакет SMTP (рис. 33). Воспользовавшись кнопкой “Capture/Forward”, проследим за маршрутом пакета от устройства к устройству.

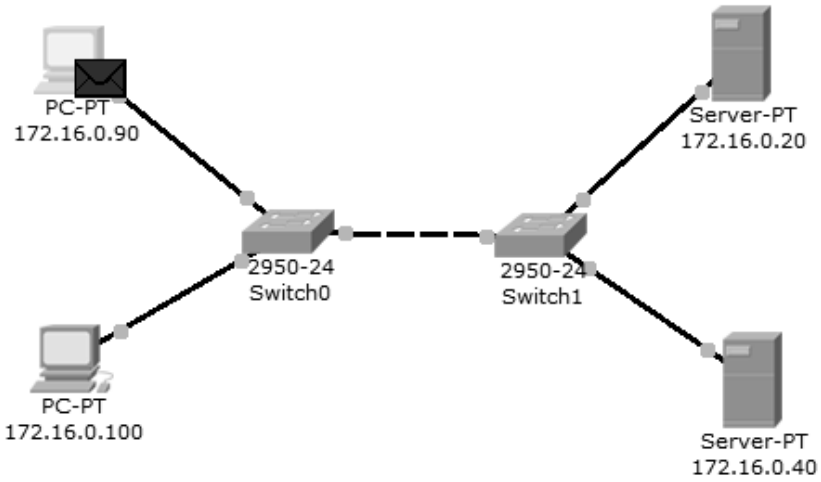
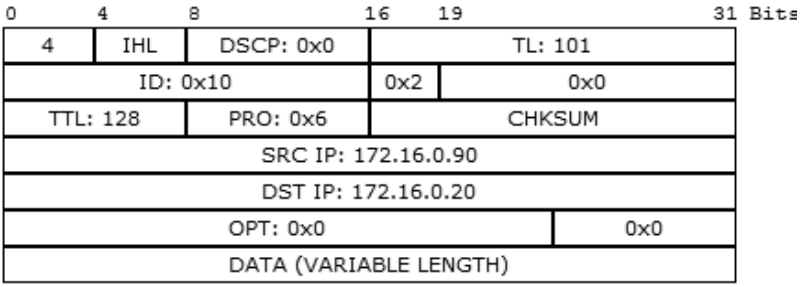


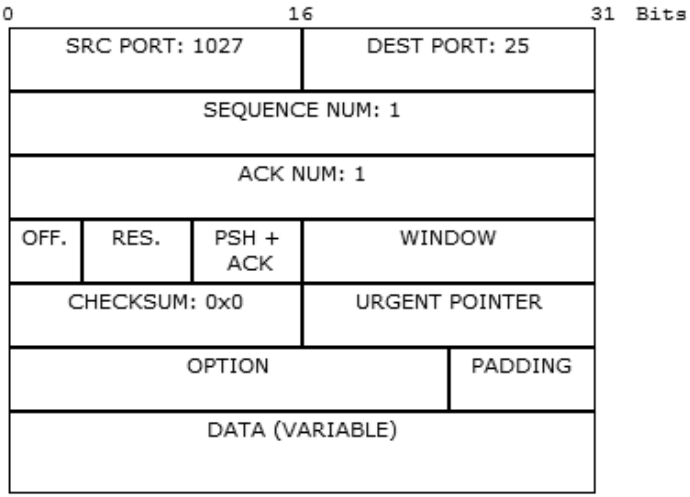
Рис. 33. Вид рабочей области

Посмотрим содержимое пакета, сформированного на узле (рис. 34).

IP



TCP



SMTP

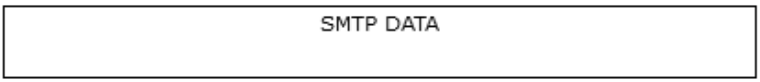


Рис. 34. Формат пакета SMTP

Пакет адресован почтовому серверу по IP-адресу 172.16.0.20. В заголовке TCP содержится порт назначения – 25. Можно сделать вывод, что пакет сформирован верно. Пакет на пути своего следования к серверу проходит через два коммутатора (рис. 35). Убедитесь, что это так.

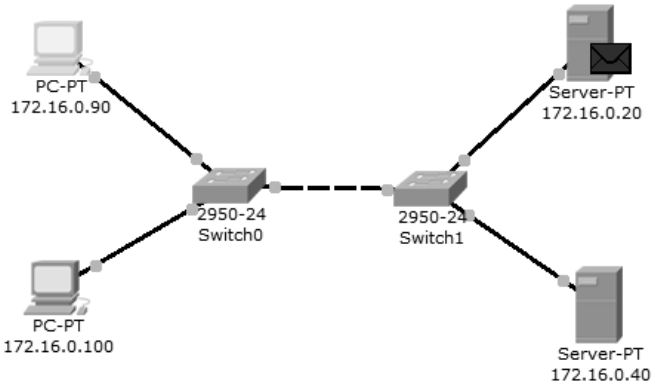


Рис. 35. Вид рабочей области

Когда пакет приходит на сервер, тот, обрабатывая его, определяет, что письмо адресовано домену *mail.ru*. Сервер 172.16.0.20 обращается к службе DNS за IP-адресом заданного сервера. По указанному адресу письмо перенаправляется на соответствующий почтовый сервер (рис. 36).

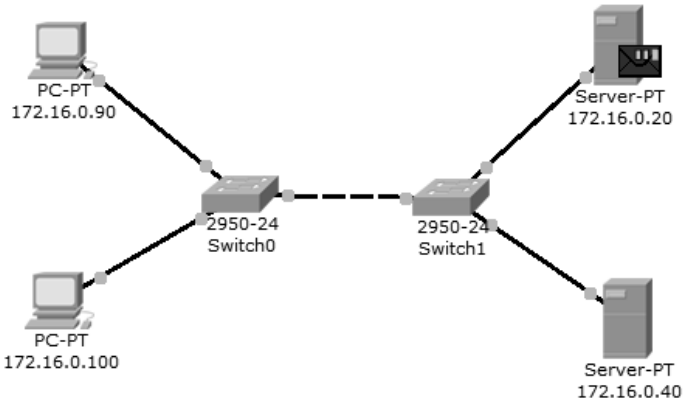


Рис. 36. Вид рабочей области

SMTP-пакет, сформированный сервером 172.16.0.20, содержит следующую информацию: IP-адрес назначения – 172.16.0.40, порт назначения – 25 (рис. 37).

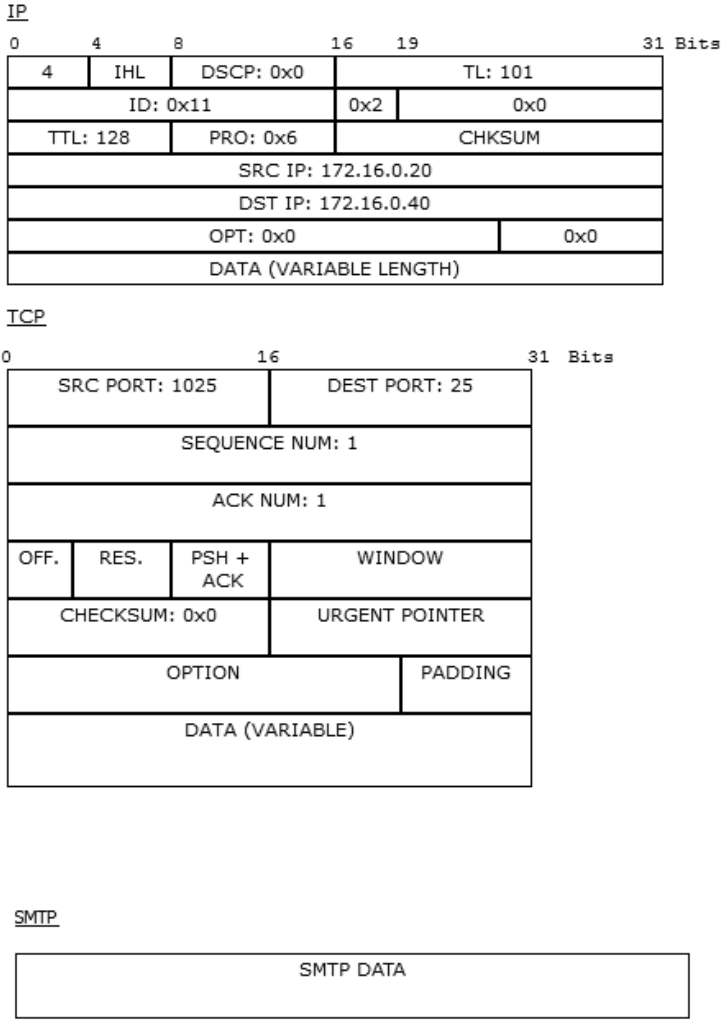


Рис. 37. Формат пакета SMTP

Пакет проходит через коммутатор Switch1 и доставляется серверу 172.16.0.40 (рис. 38).

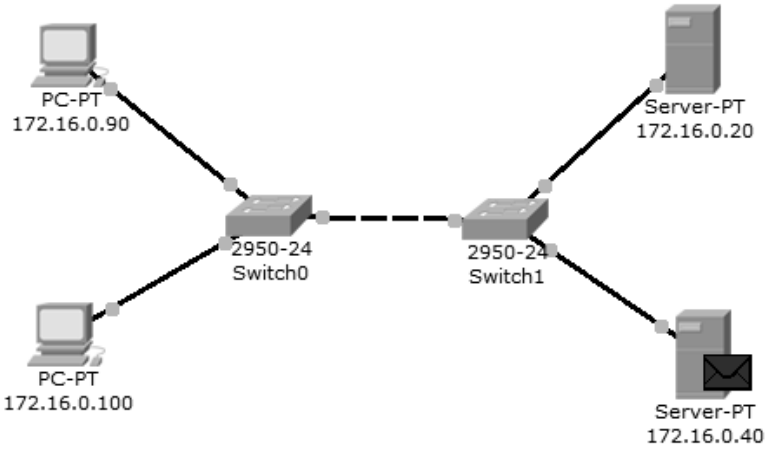


Рис. 38. Вид рабочей области

На сервере 172.16.0.40 формируется SMTP-ответ серверу 172.16.0.20 и отправляется на указанный адрес (рис. 39).

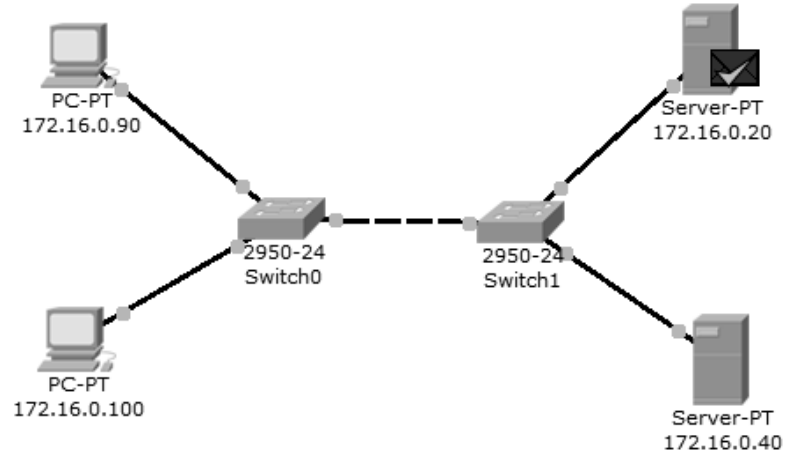


Рис. 39. Вид рабочей области

Из содержимого пакета, пришедшего обратно на сервер 172.16.0.20: IP-адрес источника – 172.16.0.40, порт источника – 25 (рис. 40).

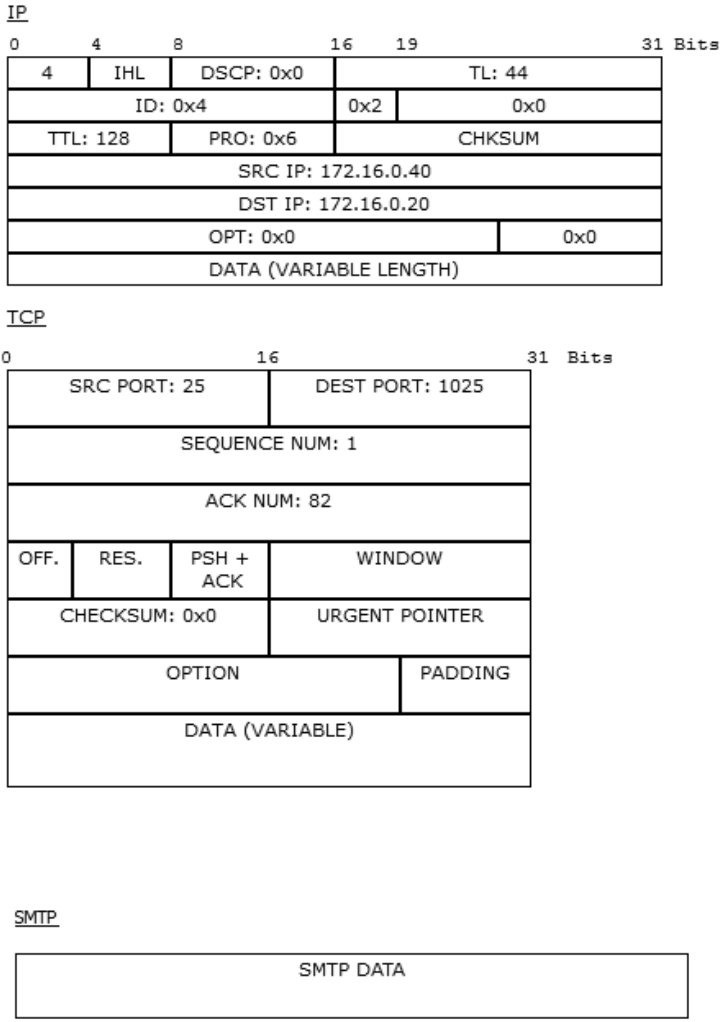


Рис. 40. Формат пакета SMTP

С помощью протокола SMTP мы отправили письмо на сервер *mail.ru*, теперь оно хранится там.

Наш адресат (узел 172.16.0.100) еще не получил отправленное письмо, так как на сервер он еще не обратился по протоколу POP3. Для получения письма необходимо проделать следующие действия:

1) Один клик по узлу 172.16.0.100.

2) Выбираем на вкладке “Desktop” программу “E-mail”.

3) Нажимаем на кнопку “Receive”, чтобы прочитать письмо.

На хосте формируется пакет протокола POP3 (рис. 41). Воспользовавшись кнопкой “Capture/Forward”, проследим за маршрутом пакета от устройства к устройству.

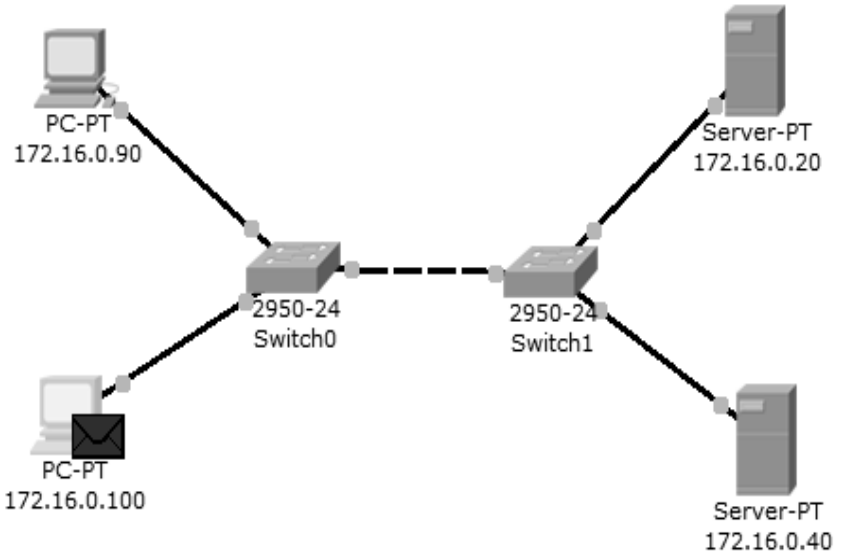
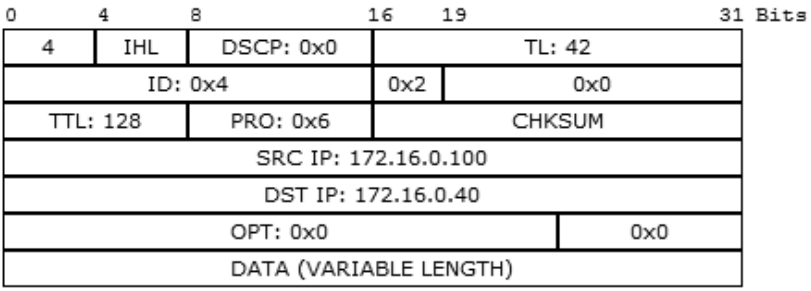


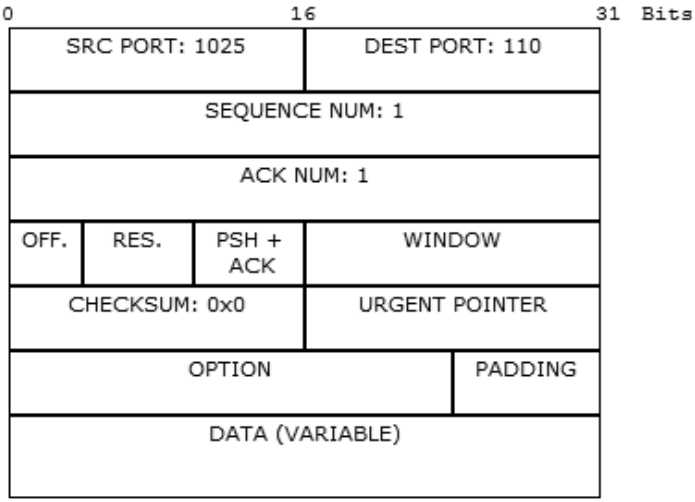
Рис. 41. Вид рабочей области

Посмотрим содержимое пакета, сформированного на узле (рис. 42).

IP



TCP



POP3

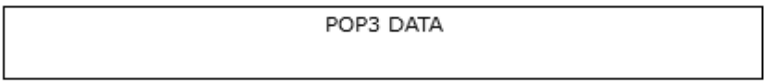


Рис. 42. Формат пакета POP3

Пакет адресован почтовому серверу по IP-адресу 172.16.0.40. В заголовке TCP содержится порт назначения – 110. Можно сделать вывод, что пакет сформирован верно. Пакет на пути своего следования к серверу проходит через два коммутатора. Убедитесь, что это так. Когда пакет приходит на сервер, тот обрабатывает его и формирует пакет-ответ (рис. 43).

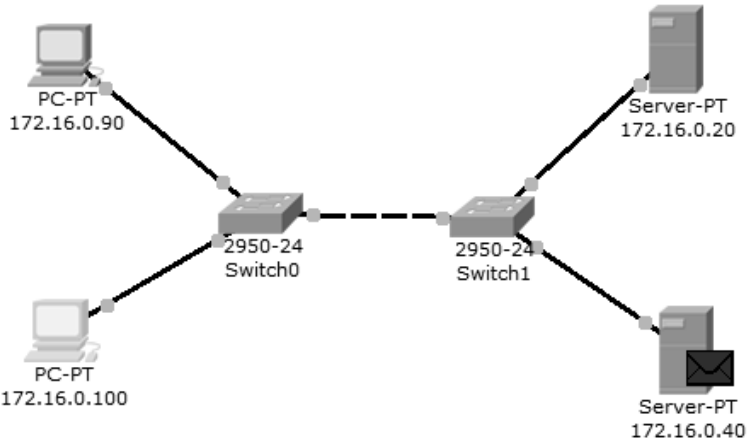


Рис. 43. Вид рабочей области

Пакет по тому же маршруту возвращается на узел 172.16.0.100 с ответом (письмом) от сервера. Посмотрим содержимое ответа (рис. 44).

Порт-источник – 110. Ответ пришел от сервера 172.16.0.40 с некоторыми POP3-данными. С помощью протокола POP3 узел 172.16.0.100 получил письмо с сервера, отправленное туда узлом 172.16.0.90 (рис. 45).

Как уже упоминалось в теоретических сведениях, почтовые протоколы SMTP и POP3 обмениваются информацией с помощью команд. Клиенту электронной почты, чтобы установить соединение с сервером/отправить письмо/разорвать соединение необходимо отправлять серверу соответствующие команды. Сервер электронной почты, в свою очередь, обрабатывает эти команды и формирует отклики для клиента. Отклики smtp-сервера содержат цифровой код ответа: успешно или с ошибкой обработана команда. Отклики pop3-сервера так же содержат два типа сообщений: успех или ошибка.

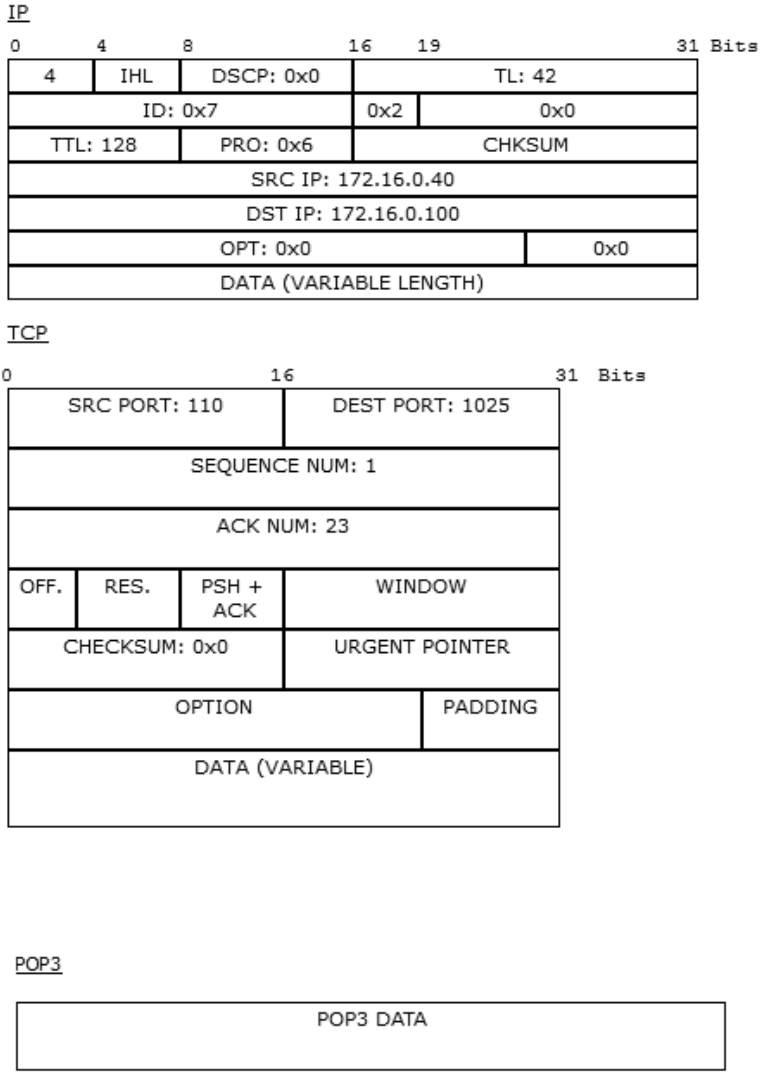


Рис. 44. Формат пакета POP3

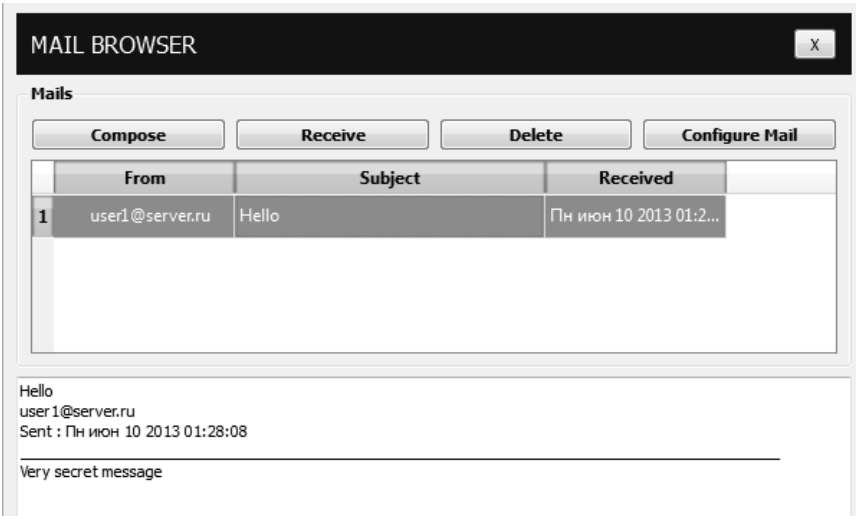


Рис. 45. Форма чтения входящих писем

Обращая внимание на содержимое пакета SMTP или POP3 протокола, видно, что на прикладном уровне пакет детально не рассматривается.

Пример приведен на рис. 46.

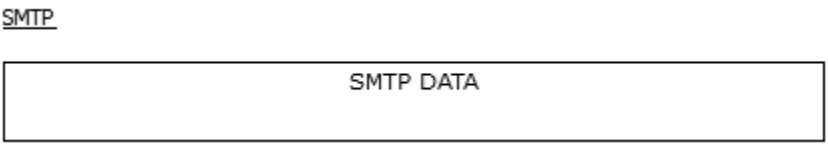


Рис. 46. Данные прикладного уровня

Поэтому эксперимент посылки письма несуществующему пользователю не является содержательным, так как подробно увидеть ответ от smtp-сервера нам не удастся. Для подробного изучения взаимодействия между клиентом и smtp- или pop3-сервером следует обратиться к предложенной спецификации RFC 2821 и RFC 1939.

ПРАКТИКУМ «АНАЛИЗ ФУНКЦИОНИРОВАНИЯ ПРОТОКОЛОВ SMTP И POP3»

Цель работы: изучить принципы организации взаимодействия прикладных программ с помощью протоколов электронной почты SMTP и POP3 в режиме симуляции Cisco Packet Tracer.

Порядок работы

1. Получение у преподавателя варианта для выполнения работы;
2. Построение топологии сети, настройка сетевых устройств в соответствии с вариантом задания;
3. Настройка почтового сервера;
4. Исследование прикладных почтовых протоколов в режиме симуляции;
5. Отправка письма по протоколу SMTP на сервер; Определение отправителя и получателя остается за обучающимся.
6. Получение письма по протоколу POP3 от сервера;
7. Оформление отчета в соответствии с требованиями;
8. Подготовка к защите выполненной работы по контрольным вопросам. Для подготовки следует отдавать предпочтение рецензируемым источникам. При использовании электронных ресурсов необходимо проверять достоверность изложенного материала.

Варианты заданий

Таблица 14

Параметры в соответствии с вариантом задания

Вариант 1			
Конечные узлы	IP-адрес	Маска сети	IP-адрес DNS-сервера
PC0	172.16.1.90	255.255.0.0	172.16.1.20
PC1	172.16.1.100	255.255.0.0	172.16.1.20
Серверы			
Server0	172.16.1.20	255.255.0.0	172.16.1.20
Server1	172.16.1.60	255.255.0.0	172.16.1.20

Продолжение табл. 14

Вариант 2			
Конечные узлы	IP-адрес	Маска сети	IP-адрес DNS-сервера
PC0	172.16.0.12	255.255.0.0	172.16.0.50
PC1	172.16.0.13	255.255.0.0	172.16.0.50
Серверы			
Server0	172.16.0.50	255.255.0.0	172.16.0.50
Server1	172.16.0.10	255.255.0.0	172.16.0.50
Вариант 3			
Конечные узлы	IP-адрес	Маска сети	IP-адрес DNS-сервера
PC0	192.168.3.1	255.255.255.0	192.168.3.8
PC1	192.168.3.3	255.255.255.0	192.168.3.8
Серверы			
Server0	192.168.3.8	255.255.255.0	192.168.3.8
Server1	192.168.3.5	255.255.255.0	192.168.3.8
Вариант 4			
Конечные узлы	IP-адрес	Маска сети	IP-адрес DNS-сервера
PC0	172.16.2.90	255.255.0.0	172.16.2.25
PC1	172.16.2.10	255.255.0.0	172.16.2.25
Серверы			
Server0	172.16.2.25	255.255.0.0	172.16.2.25
Server1	172.16.2.40	255.255.0.0	172.16.2.25
Вариант 5			
Конечные узлы	IP-адрес	Маска сети	IP-адрес DNS-сервера
PC0	192.168.5.1	255.255.255.0	192.168.5.7
PC1	192.168.5.3	255.255.255.0	192.168.5.7
Серверы			
Server0	192.168.5.7	255.255.255.0	192.168.5.7
Server1	192.168.5.5	255.255.255.0	192.168.5.7
Вариант 6			
Конечные узлы	IP-адрес	Маска сети	IP-адрес DNS-сервера
PC0	192.168.4.1	255.255.255.0	192.168.4.9
PC1	192.168.4.3	255.255.255.0	192.168.4.9

Продолжение табл. 14

Сервер			
Server0	192.168.4.9	255.255.255.0	192.168.4.9
Server1	192.168.4.6	255.255.255.0	192.168.4.9
Вариант 7			
Конечные узлы	IP-адрес	Маска сети	IP-адрес DNS-сервера
PC0	172.16.3.15	255.255.0.0	172.16.3.70
PC1	172.16.3.25	255.255.0.0	172.16.3.70
Серверы			
Server0	172.16.3.70	255.255.0.0	172.16.3.70
Server1	172.16.3.40	255.255.0.0	172.16.3.70
Вариант 8			
Конечные узлы	IP-адрес	Маска сети	IP-адрес DNS-сервера
PC0	172.16.4.90	255.255.0.0	172.16.4.30
PC1	172.16.4.10	255.255.0.0	172.16.4.30
Серверы			
Server0	172.16.4.30	255.255.0.0	172.16.4.30
Server1	172.16.4.100	255.255.0.0	172.16.4.30
Вариант 9			
Конечные узлы	IP-адрес	Маска сети	IP-адрес DNS-сервера
PC0	172.16.5.20	255.255.0.0	172.16.5.10
PC1	172.16.5.40	255.255.0.0	172.16.5.10
Серверы			
Server0	172.16.5.10	255.255.0.0	172.16.5.10
Server1	172.16.5.80	255.255.0.0	172.16.5.10
Вариант 10			
Конечные узлы	IP-адрес	Маска сети	IP-адрес DNS-сервера
PC0	172.16.6.20	255.255.0.0	172.16.6.40
PC1	172.16.6.10	255.255.0.0	172.16.6.40
Серверы			
Server0	172.16.6.40	255.255.0.0	172.16.6.40
Server1	172.16.6.30	255.255.0.0	172.16.6.40

Окончание табл. 14

Вариант 11			
Конечные узлы	IP-адрес	Маска сети	IP-адрес DNS-сервера
PC0	192.168.6.2	255.255.255.0	192.168.6.7
PC1	192.168.6.3	255.255.255.0	192.168.6.7
Серверы			
Server0	192.168.6.7	255.255.255.0	192.168.6.7
Server1	192.168.6.5	255.255.255.0	192.168.6.7
Вариант 12			
Конечные узлы	IP-адрес	Маска сети	IP-адрес DNS-сервера
PC0	192.168.7.2	255.255.255.0	192.168.7.5
PC1	192.168.7.4	255.255.255.0	192.168.7.5
Серверы			
Server0	192.168.7.5	255.255.255.0	192.168.7.5
Server1	192.168.7.8	255.255.255.0	192.168.7.5
Вариант 13			
Конечные узлы	IP-адрес	Маска сети	IP-адрес DNS-сервера
PC0	192.168.8.4	255.255.255.0	192.168.8.2
PC1	192.168.8.3	255.255.255.0	192.168.8.2
Серверы			
Server0	192.168.8.2	255.255.255.0	192.168.8.2
Server1	192.168.8.8	255.255.255.0	192.168.8.2
Вариант 14			
Конечные узлы	IP-адрес	Маска сети	IP-адрес DNS-сервера
PC0	192.168.9.3	255.255.255.0	192.168.9.6
PC1	192.168.9.4	255.255.255.0	192.168.9.6
Серверы			
Server0	192.168.9.6	255.255.255.0	192.168.9.6
Server1	192.168.9.7	255.255.255.0	192.168.9.6

Содержание отчета

1. Титульный лист.
2. Цель и задание.
3. Номер и описание варианта.
4. Процесс настройки устройств.
5. Результаты проверки корректности работы сети.
6. Анализ содержимого пакетов и объяснение полученных результатов.
7. Выводы.

Контрольные вопросы

1. Прикладной уровень модели OSI, функции и протоколы.
2. Структура стека TCP/IP, сопоставление с моделью OSI.
3. Протоколы электронной почты. Особенности функционирования, команды, сессии.
4. Назначение и основные принципы MIME.
5. Спам. Определение, назначение, способы реализации и методы защиты.

СПИСОК РЕКОМЕНДОВАННЫХ ИСТОЧНИКОВ

1. *Олифер В.Г., Олифер Н.А.* Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов. – 4-е изд. – СПб.: Питер, 2012. – 944 с.: ил.

2. *Олифер В.Г., Олифер Н.А*. Компьютерные сети. Протоколы, технологии, технологии. 3-е издание. СПб.: Изд-во Питер, 2003. - 960 с.

3. *Мулюха В.А., Новопашенный А.Г., Подгурский Ю.Е., Заборовский В.С.* Методы и средства защиты компьютерной информации. Межсетевое экранирование: учеб. пособие. – СПб.: Изд-во Политехн. университета, 2010. – 92 с.

4. *Ицыксон В.М.* Курс лекций «Технологии компьютерных сетей» 2012. [Электронный ресурс]. – Режим доступа: http://kspt.icc.spbstu.ru/2013/course/networks2

5. *Фролов А.И.* Сети ЭВМ и телекоммуникации. – Орел: 2006. – 71 с.

6. Пример сессии IMAP [Электронный ресурс]. – Режим доступа: https://studfiles.net/preview/6369835/page:16/

7. Дополнительные методы защиты электронных сообщений [Электронный ресурс]. – Режим доступа: https://studfiles.net/preview/6369835/page:16/

ОГЛАВЛЕНИЕ

Мищенко Полина Валерьевна

ОСОБЕННОСТИ ФУНКЦИОНИРОВАНИЯ ПРОТОКОЛОВ ЭЛЕКТРОННОЙ ПОЧТЫ

Учебное пособие

Выпускающий редактор *И.П. Брованова*
Дизайн обложки *А.В. Ладыжская*
Компьютерная верстка *С.И. Ткачева*

Налоговая льгота – Общероссийский классификатор продукции
Издание соответствует коду 95 3000 ОК 005-93 (ОКП)

Подписано в печать 30.11.2018. Формат 60 × 84 1/16. Бумага офсетная. Тираж 100 экз.
Уч.-изд. л. 4,18. Печ. л. 4,5. Изд. № 298. Заказ № 47. Цена договорная

Отпечатано в типографии
Новосибирского государственного технического университета
630073, г. Новосибирск, пр. К. Маркса, 20