Прикладные протоколы TCP/IP

КСПТ
КОМПЬЮТЕРНЫЕ СИСТЕМЫ И ПРОГРАММНЫЕ ТЕХНОЛОГИИ

Прикладные протоколы TCP/IP

- Удаленные терминалы
- Электронная почта
- Передача файлов
- Передача новостей
- Обмен мгновенными сообщениями
- Передача объектов
- и т.п.

1. Протоколы удаленных терминалов

- Протокол Rlogin (RFC 1282)
  - Протокол TCP, порт 513
- Протокол TELNET (RFC 854)
  - Протокол TCP, порт 23
  - Универсальный TCP-клиент
- Протокол ssh
  - Протокол TCP, порт 22
  - Обеспечивает безопасную передачу
- X-протокол (RFC 1013)
  - Графическая распределенная среда
  - Протокол UDP, порт 6000+N

2. Электронная почта

- Механизм работы почты
- Хранение почтовых сообщений
- Доставка сообщения между серверами
- Доставка сообщения клиенту

Электронная почта

- Формат почтового адреса: name@host.domain
- name:
  - Имя пользователя
  - Название почтового ящика
  - Название списка рассылки
- Формат сообщения – RFC 822
- Расширенные сообщения – MIME

Протокол SMTP

- Simple Mail Transfer Protocol
- Протокол передачи электронной почты
- Первый протокол передачи почты MTP (**RFC 772**), 1980 г.
- Первый стандарт SMTP **RFC 821**, 1982 г.
- Расширение (ESMTP) – **RFC 1651**, 1994 г.
- Действующий стандарт **RFC 2821**, 2001 г.
- Аутентификация **RFC 2554**, 1999 г.

Протокол SMTP

- Клиент <-> MTA
- MTA <-> MTA
- Использует TCP, порт 25
- Не поддерживает шифрования
- Базовая версия не поддерживает аутентификацию

Протокол SMTP. Команды

- Соединение
  - 220
  - 554
- HELO <домен>
  - 250
  - 504, 550
- MAIL FROM: <адрес>
  - 250
  - 552, 451, 452, 550, 553, 503
- RCPT TO: <адрес>
  - 250, 251
  - 550, 551, 552, 553, 450, 451, 452, 503, 550

Протокол SMTP. Команды

- DATA
  - 354

    <Текст письма>

    .
    - 250
    - 552, 554, 451, 452
  - 451, 554, 503
- NOOP
  - 250
- QUIT
  - 221

Протокол SMTP. Команды

Расширение протокола – RFC 1651 (ESMTP)

- **EHLO**
  - 250
  - 504, 550
- **VRFY** <mailbox>
  - 250, 251, 252
  - 550, 551, 553, 502, 504
- **EXPN** <maillist>
  - 250, 252
  - 550, 500, 502, 504
- **RSET** – прерывает текущую процедуру посылки почтового сообщения.
  - 250

Протокол SMTP. Аутентификация

Спецификация – RFC 2554

- Реализуется как расширение SMTP
  - Ответ на EHLO – список поддерживаемых механизмов:
    - S: 220 smtp.example.com ESMTP server ready
    - C: EHLO jgm.example.com
    - S: 250-smtp.example.com
    - S: 250 AUTH CRAM-MD5 DIGEST-MD5
- Механизмы аутентификации описаны в RFC2222 (SASL – Simple Authentication & Security Layer)
  - KERBEROS_4
  - GSSAPI
  - S/KEY

Протокол SMTP. Аутентификация

- Команда **AUTH** <механизм> [<строка>]
  - 334
  - 504, 503
- Секция команды **MAIL FROM:**
  - **MAIL FROM:** <адрес> **AUTH**=строка
  - **MAIL FROM:** <адрес> **AUTH**=<>
  - Используется для передачи идентификационной строки сообщения в «доверительных» сообществах
- POP3 before SMPT

SMTP. Маршрутизация

Цель – оптимизация почтового трафика

- **RCPT TO**: <@first.com, @second.ru, ivan@gmail.com>
- **MAIL FROM**: <@first.ru, peter@mail.ru>
- Адреса серверов перемещаются из одного списка в другой

Стандарт MIME

Multipurpose **I**nternet **M**ail **E**xtensions

Спецификация RFC 1521

Цель – передача нетекстовой информации

Кодирование:

- BASE64
- Quoted-printable

Стандарт MIME. Загловки

- Представление заголовков сообщений. Описано RFC 1522
- Формат:
  - =? Charset ? Encoding ? Encoded-text ?=
  - Charset – набор символов
  - Encoding – кодировка
    - B – BASE64
    - Q – Quoted-Printable
    - Encoded-text – текст заголовка
  - Пример:
    - Subject: Re: =?koi8-r?b?y8zVwg==?=

Стандарт MIME. Пример

```
Subject: =?KOI8-R?Q?=CB=CF=CE=F=CE=C6=C5=D2=C5=CE=C3?= =?KOI8-R?Q?=C9=C9?=
Content-Type: multipart/mixed; boundary="------------030300060608090404060101“
X-Spam-Flag: NO

This is a multi-part message in MIME format.
--------------030300060608090404060101
Content-Type: text/plain; charset=KOI8-R; format=flowed
Content-Transfer-Encoding: 8bit
Уважаемые коллеги!
******** *.*.
--------------030300060608090404060101
Content-Type: application/msword;
 name*0*=KOI8-R''%EB%CF%CE%CB%D5%D2%D3%D9%20%37%2E%30%32%2E%30%37%2E%64%6F;
 name*1*=%63
Content-Transfer-Encoding: base64
Content-Disposition: inline;
 filename*0*=KOI8-R''%EB%CF%CE%CB%D5%D2%D3%D9%20%37%2E%30%32%2E%30%37%2E%64;
 filename*1*=%6F%63

0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAAABAAAARwAAAAAA
AAAAEAAASQAAAAEAAAD+////AAAAAEYAAAD/////////////////////////////////////
.............
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==
--------------030300060608090404060101--
```

Борьба со спамом

- **SPF** (Sender Policy Framework) – 2003г.
- **Caller ID for E-mail** – от Microsoft, 2004г.
- **Sender ID** – объединение технологий:
  - RFC 4405, 4406, 4407,4408, 2006 г.
- Основная цель – не борьба со спамом, а борьба с mail spoofing

Технология Sender ID

- Идея:
  - Владелец домена публикует в открытом доступе список адресов, с которых можно отправлять почту от имени данного домена
  - Получатель почты проверяет сообщение, руководствуясь заголовками письма и записями о разрешенных именах домена
- Purported Responsible Address (PRA)— предполагаемый адрес отправителя
- Задача – извлечение из письма PRA и последующая проверка его

Технология Sender ID

- Информация о разрешенных адресах домена хранится в DNS
- Поиск PRA осуществляется по цепочкам заголовков письма
- Если PRA найден:
  - На основе PRA определяется PRD (Purported Responsible Domain)
  - Осуществляется запрос к DNS-серверу, отвечающему за домен PRD
  - Принимается решение о дальнейших действиях
- Если PRA не найден:
  - Письмо посылается на доп. проверку

Протокол POP-3

- **P**ost **O**ffice **P**rotocol, версия **3**
- Протокол доступа к почтовому ящику
- Первый стандарт **RFC 1081**, 1988 г.
- Действующий стандарт **RFC 1939**, 1996 г.
- Заменил протокол POP-2

Протокол POP-3

- Доступ к удаленному почтовому ящику
- Аутентификация пользователя
- Просмотр списка писем
- Копирование писем в локальный ящик
- Удаление писем с сервера
- Ящик представляется одной почтовой папкой (Inbox)
- Используется протокол TCP, порт 110

Протокол POP-3. Команды

- **USER** <имя>
  - **-ERR** – если не поддерживается plaintext authentication
  - **+OK** – в остальных случаях
- **PASS** <пароль>
  - **+OK** – в случае успешной аутентификации
  - **-ERR** – в случае неуспешной аутентификации
- **STAT**
  - **+OK** <N> <M> – N писем общей длиной M
- **LIST**
  - **+OK**
    1 <длина 1>
    N <длина N>

Протокол POP-3. Команды

- **LIST** <N>
  - **+OK**
    <N> <длина>
  - **-ERR** – сообщение отсутствует
- **RETR** <number>
  - **+OK**
    <текст сообщения>
    .
  - **-ERR** – сообщение отсутствует
- **DELE** <N>
  - **+OK**
  - **-ERR** – сообщение отсутствует

Протокол POP-3. Команды

- **RSET**
  - **+OK**
- **QUIT**
  - **+OK**
- **TOP** <N> <M>
  - **+OK**
    <заголовок сообщения N>
    <пустая строка>
    <M строк сообщения>
  - **-ERR** – сообщение отсутствует

Протокол POP-3. Команды

- **UIDL** <N>
  - **+OK** <N> <UID>
  - **-ERR** – сообщение отсутствует
- **UIDL**
  - **+OK**
    1 <UID 1>
    2 <UID 2>

Протокол POP-3. Аутентификация

- **APOP** <имя> <дайджест>
  - **+OK**
  - **-ERR** – ошибочная аутентификация

Дайджест вычисляется по алгоритму MD5 (RFC1231)

В вычислении используется строка сервера <pid.clock@hostname>

- **AUTH** <тип_аутентификации> (RFC 1734)
  - **-ERR** – неизвестный метод аутентификации
  - + – ответы сервера
  - **+OK**

  Формат RFC 1731 “IMAP4 Authentication Mechanisms”

Протокол POP-3. Резюме

Недостатки

- отсутствие шифрования
- аутентификация
- блокировка ящика
- отсутствие папок
- отсутствие атрибутов сообщений

Достоинства

- простота реализации
- большая распространенность

Протокол IMAP-4

- Internet Mail Access Protocol, версия 4
- Первый стандарт RFC 1730, 1994 г.
- Действующий стандарт **RFC 2060** 1996 г.
- Создан как альтернатива POP-3

IMAP-4. Особенности

- Позволяет хранить удаленную структуру папок сообщений
- Обеспечивает асинхронный обмен командами
- Уникальный номер команды и ответа
- Флаги сообщений
- Уникальные идентификаторы сообщений
- Механизмы копирования и перемещения сообщений
- Средства поиска сообщений
- Варианты аутентификации (login и authenticate)
- Использует протокол TCP, порт №143

IMAP-4. Флаги сообщений

- Системные флаги
  - \Seen
  - \Answered
  - \Deleted
  - \Draft
  - \Recent
- Пользовательские флаги

IMAP-4. Команды

- На всех стадиях
  - **CAPABILITY** – запрос списка возможностей
  - **NOOP**
  - **LOGOUT**
- Стадия «Неаутентифицирован»
  - **LOGIN** <username> <password>
  - **AUTHENTICATE** <method>

IMAP-4. Команды

- На стадия «Аутентифицирован»
  - **SELECT** <имя_ящика> – выбор ящика
  - **EXAMINE** <имя_ящика> – выбор ящика (RO)
  - **CREATE** <имя_ящика> – создание ящика
  - **DELETE** <имя_ящика> – удаление ящика
  - **RENAME** <старое_имя> <новое_имя> – переименование ящика
  - **SUBSCRIBE** <имя_ящика> – подписка на ящик
  - **UNSUBSCRIBE** <имя_ящика> – отмена подписки на ящик

IMAP-4. Команды

- На стадия «Аутентифицирован»
  - **LIST** <база> <имя_ящика> – выдача списка ящиков
  - **LSUB** – <база> <имя_ящика> – выдача списка подписанных ящиков
  - **STATUS** <имя_ящика> [<имена_эл-тов_состояния>] – выдача состояния ящика
  - **APPEND** <имя_ящика> [(флаги)] [...] – добавить сообщение в ящик

IMAP-4. Команды

- На стадия «Выбран»
  - **CHECK** – проверка ящика
  - **CLOSE** – удаление помеченных сообщений и закрытие ящика
  - **EXPUNGE** – удаление помеченных сообщений
  - **SEARCH** [CHARSET] <критерии> – поиск сообщения
  - **FETCH** <набор_сообщений> <эл-ты данных>
  - **STORE** <набор_сообщений> <значение> – изменение флагов сообщений
  - **COPY** <набор_сообщений> <имя_ящика> – копирование сообщений в ящик
  - **UID** <команда> – выдача идентификаторов сообщений

IMAP–4. Отклики

- SEARCH
- FLAGS
- EXISTS
- RECENT
- EXPUNGE
- FETCH

- OK
- BAD
- PREAUTH
- BYE
- CAPABLITY
- LIST
- LSUB
- STATUS

3. Протоколы передачи файлов

- Протокол sftp (RFC 913)
  - Протокол TCP, порт 115
- Протокол tftp (RFC 1350)
  - Протокол UDP, порт 69
- Протокол ftp (RFC 959)
  - Протокол TCP, порты 21 и 20*

Протокол FTP

- File Transfer Protocol
- Первый стандарт **RFC 114**, 1971 г.
- Действующий стандарт **RFC 959**, 1985 г.
- Один из базовых протоколов TCP/IP
- Использует транспорт TCP
- Поддерживает два режима передачи

Протокол FTP

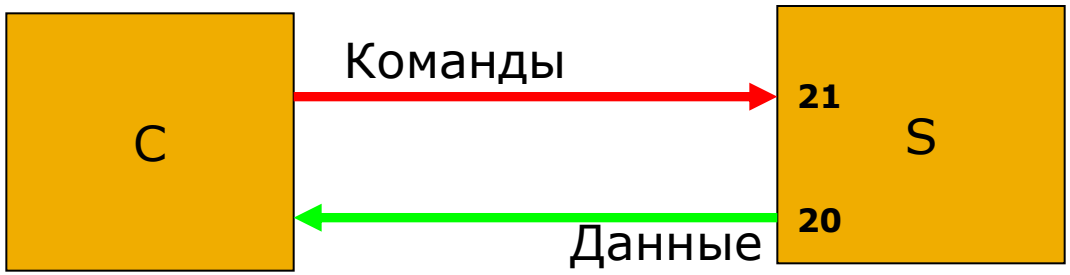
Двухканальная схема передачи:

- Управляющий канал
  - Предназначен для передачи команд
  - Существует все время обмена
  - Используется TCP-порт №21
- Канал данных
  - Предназначен для передачи файлов и каталогов
  - Организуется на время передачи
  - Используется TCP-порт №20 или непривилегированный порт

FTP. Активный режим

- Режим «по умолчанию»
- Сервер инициирует соединение данных
- Клиент открывает слушающий порт
- Номер TCP-порта сервера – 20
- Невозможно использовать с технологиями типа NAT, Proxy
- Обычно запрещён в межсетевых экранах

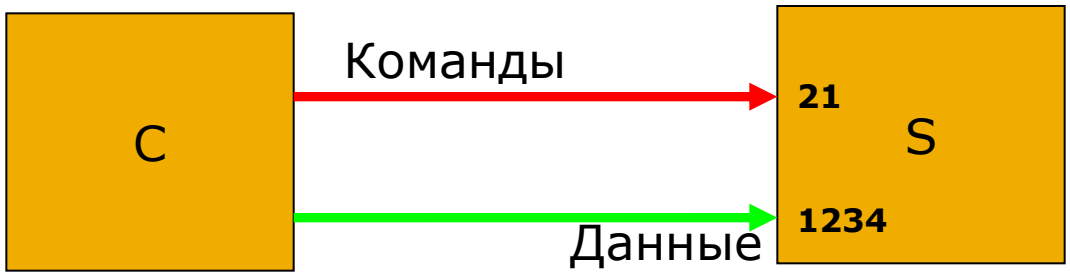
FTP. Активный режим



FTP. Пассивный режим

- Клиент инициирует соединение данных
- Сервер информирует о параметрах канала данных
- Сервер открывает слушающий порт
- Поддерживается не всеми реализациями

FTP. Пассивный режим



Протокол FTP. Команды

- **USER** <имя>
- **PASS** <пароль>
- **REIN** – реинициализация
- **ABOR** – прервать обмены
- **QUIT**

Протокол FTP. Команды

Команды, оперирующие с файловой системой

- **DELE** <имя>- удалить файла
- **RNFR** <имя> - переименовать из
- **RNTO** <имя> - переименовать в
- **CWD** <путь> - сменить каталог
- **CDUP** - перейти в родительский каталог
- **RMD** <имя> - удалить каталог
- **MKD** <имя> - создать каталог
- **PWD** - показать текущий каталог

КСПТ
КОМПЬЮТЕРНЫЕ СИСТЕМЫ И ПРОГРАММНЫЕ ТЕХНОЛОГИИ

Протокол FTP. Команды

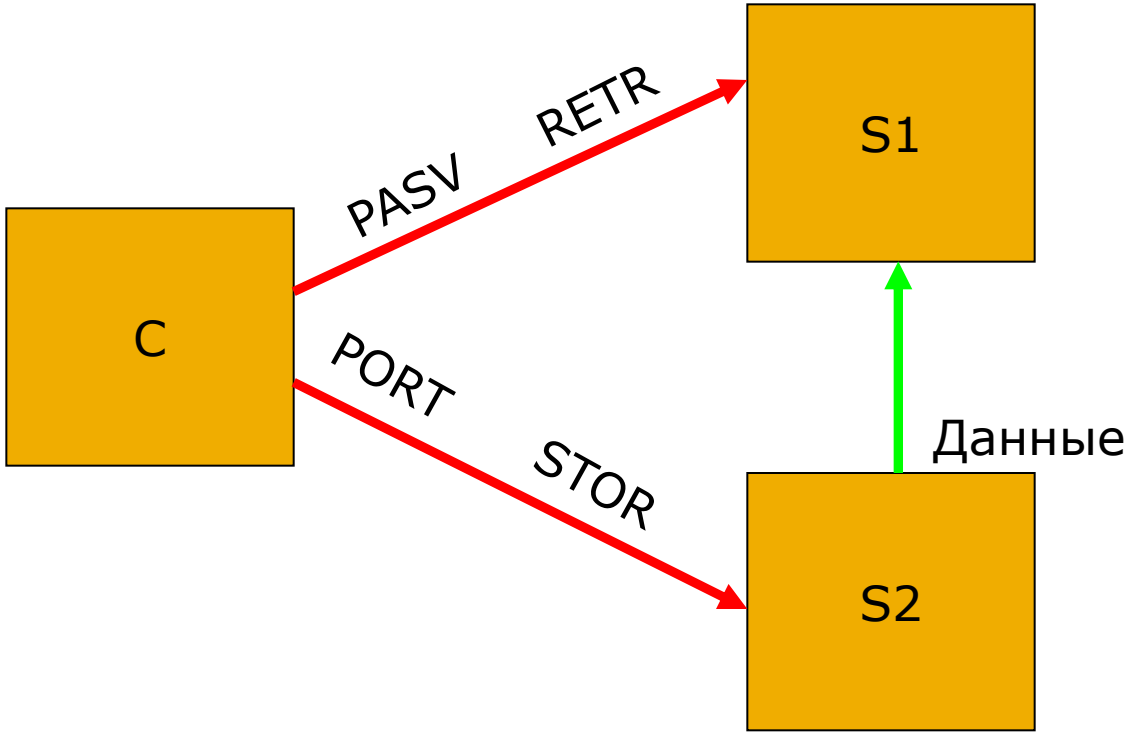
- **PORT** a1, a2, a3, a4, p1, p2 – перевод сервера в активный режим
  Address = ‘a1.a2.a3.a4’
  Port = p1*256+p2
- **PASV** – перевод сервера в пассивный режим
  - 227 a1, a2, a3, a4, p1, p2
- **TYPE** {A|E|I} – представление информации
  - A – ASCII
  - E – EDCDIC
  - I – Image
- **MODE** {S|B|C} – режим передачи данных
  - S – Stream
  - B – Block
  - C – Compressed

Протокол FTP. Команды

Команды, использующие канал данных

- **RETR** <имя> – получить файл
- **STOR** <имя> – записать файл
- **LIST** [<путь>] – получить список файлов с атрибутами
- **NLST** [<путь>] – получить список имен файлов

Использование FTP



Протокол FTP. Резюме

Недостатки

- Не поддерживает шифрования
- Не поддерживает безопасной аутентификации
- Не поддерживает современных средств адресации
- Сложность работы с защищенными сетями

Достоинства

- Эффективность
- Гибкость

Протокол HTTP

- Протокол передачи файлов
- Протокол передачи объектов
- HTTP/1.0 RFC 1945, 1996 г.
- HTTP/1.1 RFC 2068, 1997 г.
- Действующий стандарт HTTP/1.1 RFC 2616, 1999г.

Формат HTTP-запроса

<Request-line> – строка запроса
<General-header> – общий заголовок
<Request-header> – заголовок запроса
<Entity-header> – заголовок сообщения

<Body> – тело

Протокол HTTP. Строка запроса

Формат: <METHOD> <URL> <HTTP-VERSION>

Методы:

- GET
- POST
- HEAD
- PUT
- DELETE
- OPTIONS
- и т.п.

Версия: HTTP/1.0 или HTTP/1.1

Протокол HTTP. Заголовки

Общий заголовок (General-header)
Присутствует, когда есть тело сообщения

- Connection:
- Data:
- Pragma:
- Transfer-encoding:
- Upgrade:
- no-cache:
- И т.д.

Протокол HTTP. Заголовки

Заголовок запроса (Request-header)

- Accept: принимаемый контент
- Accept-Charset: принимаемый набор символов
- Accept-Encoding: compress, zip
- Accept-Language: da, ru
- Authorization: basic xxx=******
- From:
- Host:
- If-modified-since:…
- Referer:
- User-agent:
- И т.д.

Протокол HTTP. Заголовки

Заголовок сообщения (Entity-header)

- Allow: GET, POST, HEAD
- Content-Encoding: x-zip
- Content-Language:
- Content-Length: 1245
- Content-Type: ...text/html; charset=win-1251
- Expires:
- Last-Modified:

Протокол HTTP. Формат ответа

<Status-line> – Строка статуса
<General-header> – общий заголовок
<Response-header> – заголовок ответа
<Entity-header> – Заголовок сообщения

<Body> – тело

Протокол HTTP. Строка статуса

Формат: <HTTP-VERSION> <Code> <Phrase>

Code:

- 1xx – информационные
- 2xx – OK
- 3xx – Переадресация (redirection)
- 4xx – Ошибка клиента
- 5xx – Ошибка сервера

Протокол HTTP. Заголовки

Заголовок ответа (Response-header)

- Location: переадресация
- Server: спецификация сервера
- WWW-Authenticate: basic realm=’localzone’
- Age: Возраст ресурса

Развитие протокола HTTP

- Протокол SPDY
  - Разработан в 2009 году компанией Google
  - Цель – эффективная замена HTTP
  - Обеспечивается сжатие, мультиплексированная передача и т.п.
  - с 2015 года не поддерживается
- Протокол HTTP/2
  - Создан на основе SPDY в 2015 году
  - Стандарт RFC 7540 (середина 2015 года)
  - Основные возможности
    - Сжатие данных и заголовков
    - Мультиплексирование запросов в одном TCP-соединении
    - Посылка ответов без запросов (в т.ч., опережающая посылка)
    - Push-уведомления
    - Приоритезация запросов
    - Безопасность
  - Поддерживается всеми браузерами и веб-серверами
  - Около 12% сайтов поддерживают HTTP/2