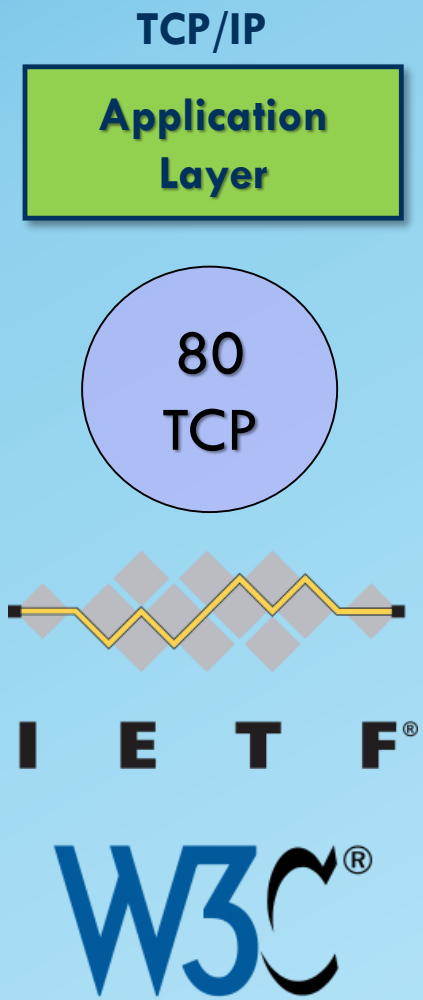
Лекция 14

Прикладные службы TCP/IP

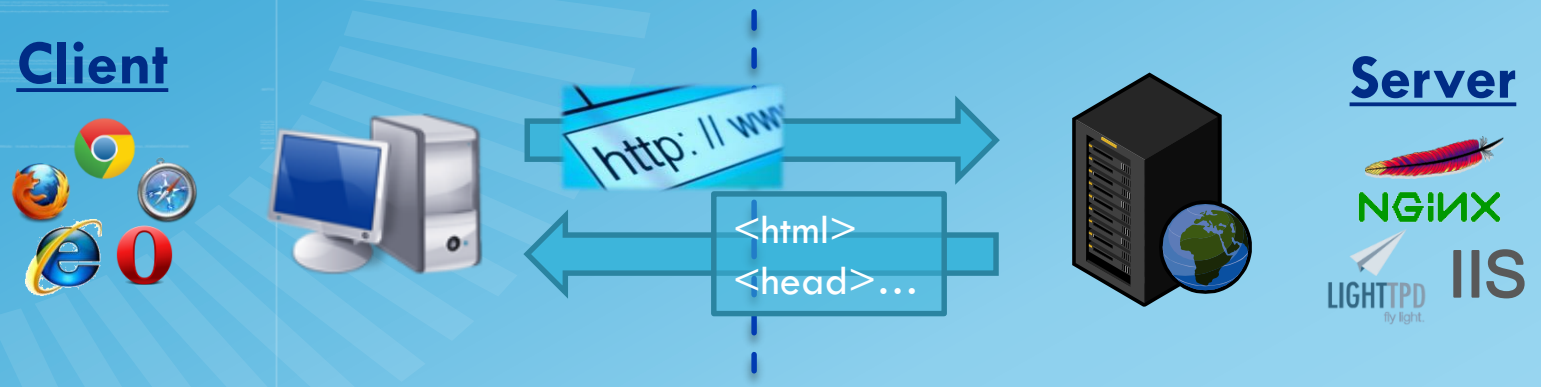
HTTP, HTTPS, Электронная почт@

Протокол HTTP

- **HTTP** (HyperText Transfer Protocol – «протокол передачи гипертекста») – клиент-серверный протокол передачи данных прикладного уровня (в первую очередь в виде гипертекстовых сообщений)
- Основной стандарт – RFC 2616 (1999) – **Hypertext Transfer Protocol -- HTTP/1.1**, разработан под руководством **IETF** (Internet Engineering Task Force) и **W3C** (World Wide Web Consortium)



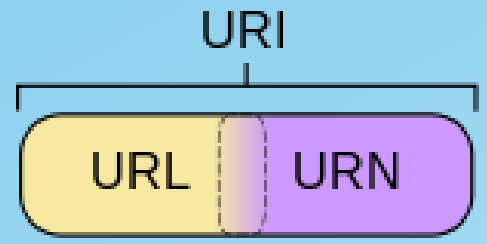
Клиент-серверная архитектура



- Веб-браузер (клиент) посылает на сервер запрос на получение ресурса (например, веб-страницы), идентифицируемого по URI
- Веб-сервер формирует ответ из статуса обработки запроса и запрошенного ресурса в оговоренном формате

Uniform Resource Identifier – URI

- **Uniform Resource Identifier (URI)** – текстовая строка, идентифицирующая ресурс (документ, изображение, файл, службу, ящик электронной почты), как правило, в сети Интернет
- Основной стандарт: RFC 3305
- Пространство URI включает
  - имена ресурсов – URN (uniform resource name)
  - локаторы ресурсов – URL (uniform resource locator)
- Синтаксис URI – *схема:идентификатор*




```
http://www.stankin.ru
mailto:ss@stankin.ru
file://C:\Windows\system32\drivers\etc\hosts
urn:isbn:0-486-27557-4
```

Uniform Resource Locator – URL

- **Uniform Resource Locator (URL)** – стандартизированный способ записи адреса ресурса в сети Интернет
- Является подмножеством URI (хотя часто ошибочно употребляется как синоним URI)
- Синтаксис URL:

  схема :// *логин* : *пароль* @ *хост* : *порт* /
  путь ? *запрос* # *фрагмент*

```
http://www.stankin.ru:8080/index.php?cat=12
http://www.stankin.ru/news/index.html#top
ftp://admin:qwerty@ftp.stankin.ru/upload/1.rar
```

HTTP-запрос

- **Стартовая строка** (Request line) – определяет ресурс и действие над ним
  - метод URI HTTP/версия
- **Заголовки** (Headers) – определяют дополнительные параметры запроса (обязательный параметр **Host**)
- **Пустая строка** – спецсимволы <CR><LF> – carriage return, line feed
- (опционально) **Тело сообщения**

```
GET /index.html HTTP/1.1
Host: www.example.com

```

Основные методы HTTP/1.1 (1)

- Условно «безопасные» методы (не меняют содержимого сервера):
  - **GET*** – загружает с сервера копию ресурса
  - **HEAD*** – аналог метода GET, ответ не содержит тела ресурса (только метаинформация)
  - **OPTIONS** – показывает разрешенные сервером методы для указанного URL
  - **TRACE** – сервер пересылает полученный запрос обратно (контроль изменений запроса промежуточными серверами)

* – методы обязательны к поддержке всеми веб-серверами

Основные методы HTTP/1.1 (2)

- «Небезопасные» методы (могут повлиять на работу сервера):
  - **POST** – передает на сервер данные для обработки (например, из веб-формы) – создает новый ресурс или изменяет существующий
  - **PUT** – загружает копию ресурса на сервер
  - **DELETE** – удаляет ресурс с сервера
  - **CONNECT** – преобразует соединение в прозрачный TCP/IP туннель (например, для HTTPS через незащищенный прокси-сервер)
  - **PATCH** – частично модифицирует ресурс

HTTP-ответ

- **Стартовая строка** – определяет тип ответа
  - HTTP/версия КодСостояния Пояснение
- **Заголовки** (Headers) – определяют дополнительные параметры запроса
- **Тело сообщения** – запрошенный ресурс

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Etag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Content-Length: 438
Connection: close
Content-Type: text/html; charset=UTF-8
<html>…</html>
```

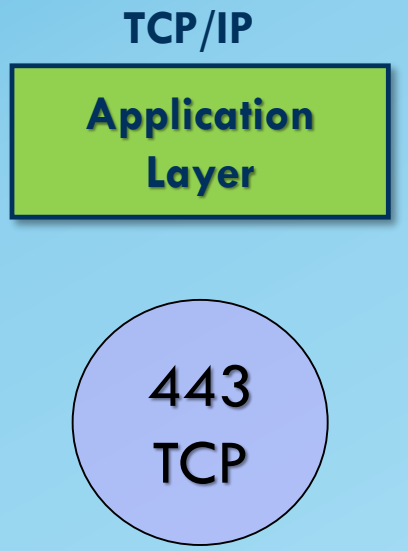
Коды состояний HTTP/1.1

- **Код состояния** (Status Code) – трехзначное число, определяющее тип ответа HTTP
  - 1xx – информационные сообщения
  - 2xx – успешное выполнение операции
  - 3xx – перенаправление
  - 4xx – ошибка на стороне клиента
  - 5xx – ошибка на стороне сервера

```
200 OK
404 Page Not Found
500 Internal Server Error
```

Протокол HTTP Secure (https)

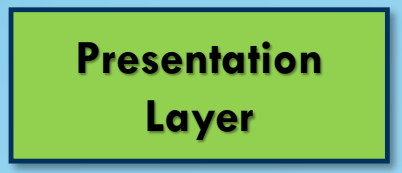
- **HTTPS** — расширение функций протокола HTTP с помощью протоколов шифрования TLS/SSL, обеспечивающее зашифрованную передачу данных и безопасную идентификацию веб-сервера
- Основной стандарт – RFC 2616 (2000) – **HTTP over TLS**



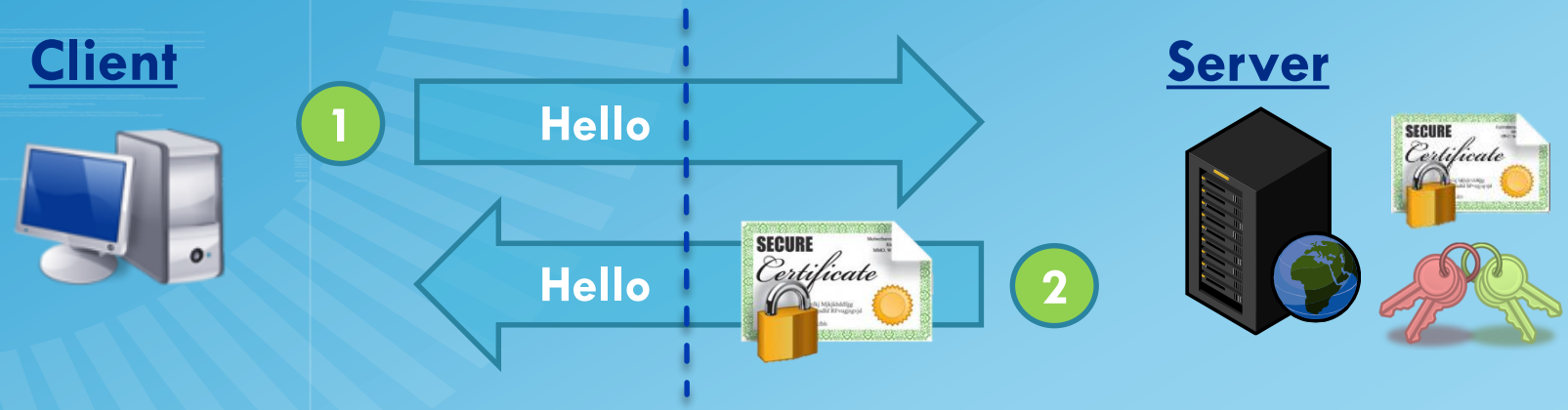
TLS/SSL

- **Transport Layer Security** (TLS) и его предшественник **Secure Sockets Layer** (SSL) – асимметричные криптографические протоколы, обеспечивающие безопасную передачу данных по сети Интернет (www, e-mail, VoIP и др.)
- Основной стандарт – RFC 5246 – **The Transport Layer Security (TLS) Protocol Version 1.2**

OSI

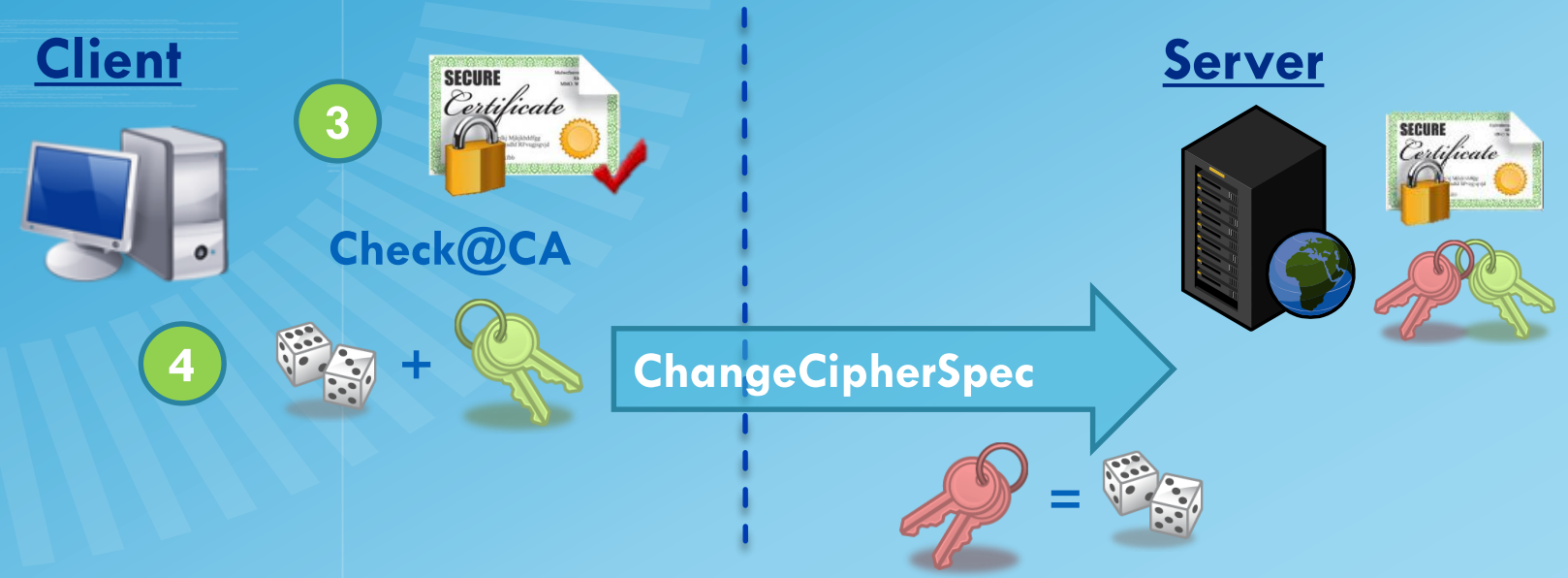


TLS – принцип работы (1)



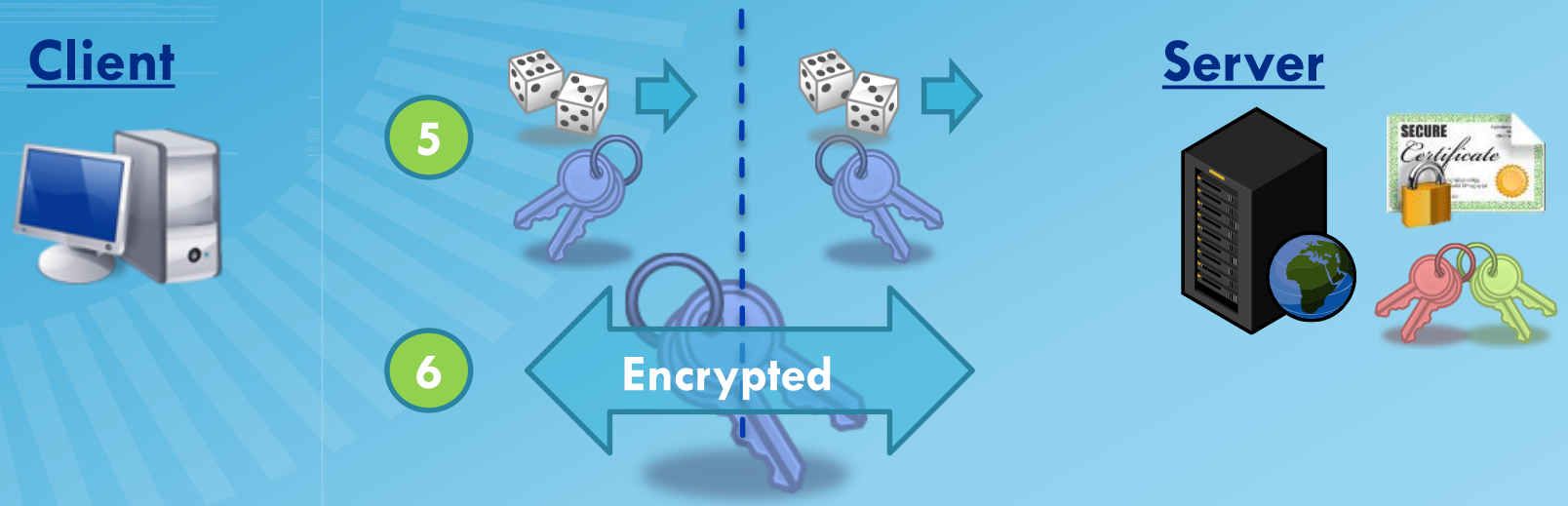
- **Клиент** инициирует соединение, передает серверу перечень поддерживаемых алгоритмов шифрования и хеширования
- **Сервер** выбирает наиболее стойкий алгоритм, сообщает клиенту и пересылает цифровой сертификат подлинности, содержащий:
  - имя сервера (server name)
  - имя удостоверяющего центра (certificate authority, CA)
  - открытый ключ сервера (public key)

TLS – принцип работы (2)



- **Клиент** проверяет сертификат в CA, генерирует случайное число, шифрует открытым ключом сервера и передает серверу
- Только **сервер** может расшифровать полученное случайное число своим закрытым ключом

TLS – принцип работы (3)



- На основе указанного числа **клиент** и **сервер** генерируют ключ сессии
- Ключ сессии используется для дальнейшего симметричного шифрования передаваемых данных

Электронная почта (e-mail)

- **Электронная почта** – технология и сервис приёма и передачи электронных сообщений по сети

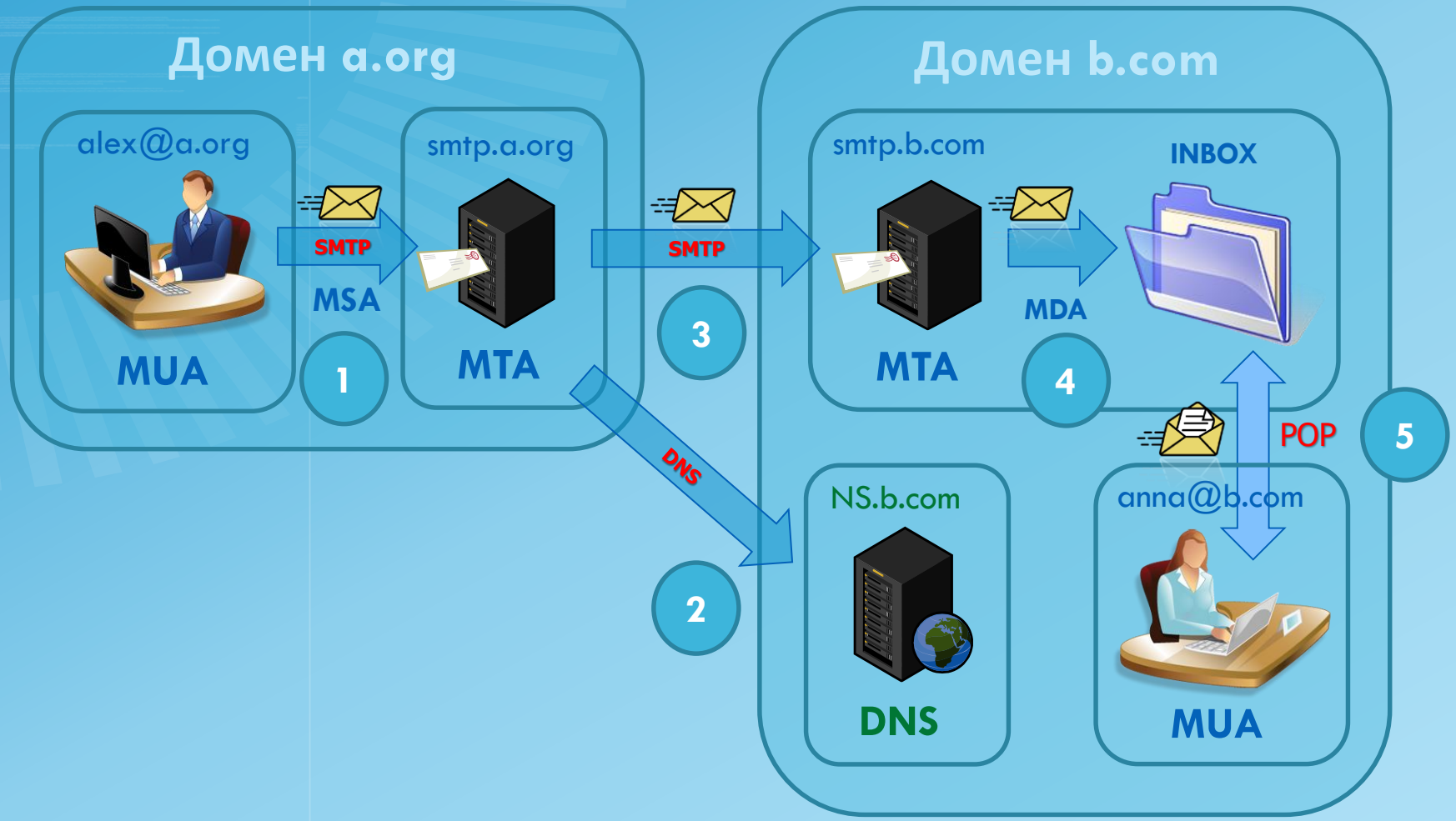
Достоинства:

- простота и доступность сервиса
- высокая надежность
- удобные адреса
- возможность пересылки текста и файлов любого типа

Недостатки:

- спам
- отсутствие гарантий доставки сообщений
- возможность задержки при пересылке сообщений

Электронная почта



Роли: MUA, MSA

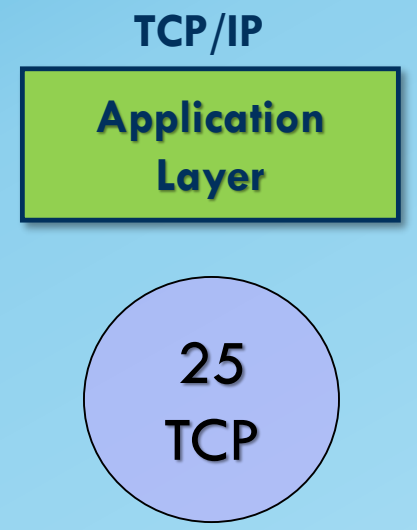
- **MUA** (Mail User Agent, пользовательский почтовый агент) – клиентское программное обеспечение для работы с электронной почтой (создания, отправки, получения и просмотра сообщений)
  - «толстый» клиент (Outlook, The Bat и др.)
  - «тонкий» веб-клиент
- **MSA** (Mail Submission Agent) – программное обеспечение, принимающее созданные сообщения у MUA и передающее их MTA для пересылки (сервер исходящей почты) – часто интегрирован с MTA

Роли: MTA, MDA

- **MTA** (Mail Transfer Agent, агент пересылки почты) – программное обеспечение для пересылки электронной почты между компьютерами (клиентами и серверами)
  - клиентская часть – отправка
  - серверная часть – получение
- **MDA** (Mail Delivery Agent, агент доставки почты) – программное обеспечение, принимающее входящие сообщения у MTA и распределяющее их по почтовым ящикам адресатов (mailbox'ам)

Протокол SMTP

- **SMTP** (Simple Mail Transfer Protocol) – клиент-серверный сетевой протокол, для передачи электронной почты в сетях TCP/IP
  - от клиента к серверу
  - между серверами
- Основной стандарт – RFC 5321 (2008) – Simple Mail Transfer Protocol



Основные команды SMTP / пример

- **HELO [username]** – идентифицирует SMTP-сервер отправителя, открывает сеанс
- **MAIL FROM <username>**
- **RCPT TO <username>**
- **DATA <...>** – указывает на начало сообщения (для окончания сообщения указывается точка)
- **QUIT** – завершает SMTP-сеанс

```
S: (ожидает соединения – слушает 25 порт TCP)
C: (подключается к порту 25 сервера)
```

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

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

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

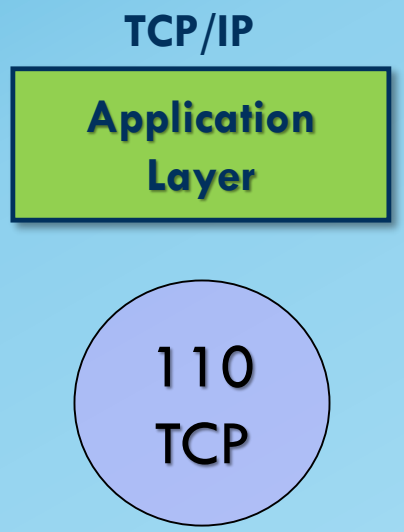
```
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: This is a test message.
C: .
S: 250 2.0.0 l3PDY91f000484 Message accepted for
delivery
```

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

```
S: (закрывает соединение)
```

Протокол POP3

- **POP3** (Post Office Protocol v.3) – стандартный клиент-серверный сетевой протокол, для получения электронной почты в сетях TCP/IP по принципу «загрузи-и-удали»
  - от клиента (MUA) к серверу
- Основной стандарт – RFC 1939 (1996) – Post Office Protocol – Version 3



POP3 состояния сеанса

- **Авторизация** – клиент проходит процедуру аутентификации, ящик блокируется
- **Транзакция** – клиент получает информацию о состоянии почтового ящика, принимает и удаляет почту
- **Обновление** – сервер удаляет выбранные письма и закрывает соединение

Основные команды POP3 (1)

- **USER <username>** – передаёт серверу имя пользователя
- **PASS <password>** – передаёт серверу пароль (PLAINTEXT!)
- **APOP <username> <digest>** – передаёт серверу имя и digest (хэш временной метки и пароля)
- **STAT** – сервер возвращает количество сообщений в почтовом ящике плюс размер почтового ящика

```
S: (ожидает соединения – слушает 110 порт TCP)
C: (Подключается к порту 110 сервера)
S: +OK pop3 server ready
```

```
C: USER mrose
S: +OK user accepted
C: PASS mypassword
S: +OK pass accepted
```

```
C: APOP mrose
c4c9334bac560ecc979e580
01b3e22fb
S: +OK
```

```
S: mrose's maildrop has 2 messages (320 octets)
```

```
C: STAT
S: +OK 2 320
```

Основные команды POP3 (2)

- **LIST** [msg id] – запрос информации о сообщении (или обо всех – без аргумента)
- **RETR** <msg id> – сервер передаёт сообщение с указанным номером
- **DELE** <msg id> – сервер помечает указанное сообщение для удаления
- **QUIT** – завершение сеанса

```
C: LIST
S: +OK 2 messages (320 octets)
S: 1 120
S: 2 200
S: .
```

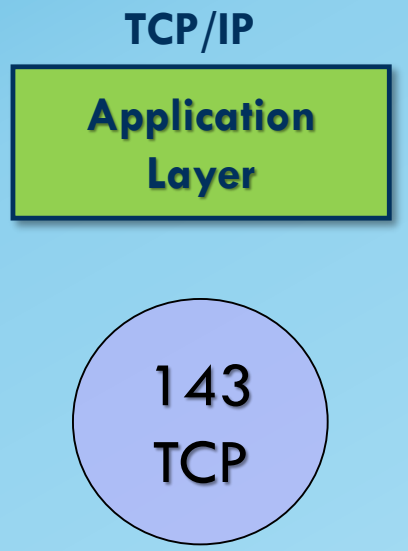
```
C: RETR 1
S: +OK 120 octets
S: <Передаёт сообщение 1>
S: .
```

```
C: DELE 1
S: +OK message 1 deleted
```

```
C: QUIT
S: +OK
C: <закрывает соединение>
S: <продолждает ждать входящие соединения>
```

Протокол IMAP

- **IMAP** (Internet Message Access Protocol) – стандартный клиент-серверный сетевой протокол, для полного доступа к электронной почте на сервере в сетях TCP/IP
- Основной стандарт – RFC 3501 (2003) – Internet Message Access Protocol – Version 4rev1



Особенности IMAP

- Не разрывает соединение, пока активен пользовательский интерфейс клиента
- Разрешает одновременное подключение нескольких клиентов к одному ящику
- Хранит состояния сообщения на сервере (флаги: прочитано, отправлено и т.д.)
- Каталогизация, перемещение и редактирование сообщений на сервере
- Поиск сообщений на стороне сервера