

## Лекция 6. Протоколы UDP, TCP, FTP, HTTP и другие

### Транспортный уровень

Взаимодействие процессов — идентификация — номер порта  
 /etc/services — номера портов действующих служб  
 до 255 стандартизованы, до 1023 большинство фактически занято

UDP — User Datagram Protocol  
 TCP — Transmission Control Protocol

**UDP**, рассылка сообщений без подтверждения

битов	назначение	примечание
16	порт отправителя	
16	порт получателя	
16	длина всего пакета в байтах	
16	UDP контрольная сумма	+ SA + DA + протокол + длина (из IP)
...	данные	

**TCP**, сеанс обмена данными с надежной доставкой

посылка подтверждений  
 таймеры  
 буферы отправки и получения  
 повторные сообщения  
 фрагментация, сборка, уничтожение дубликатов  
 контроль соединения

Процедура установления и разрыва соединения  
 TCB, Transmission Control Block  
 создается для каждого соединения

**TCP формат пакета**

битов	назначение	примечание
16	порт отправителя	
16	порт получателя	
32	SN, sequence number	смещение фрагмента
32	ACK N, acknowledgment number	смещение ожидаемого фрагмента (подтверждение)
4	hlen, длина заголовка	в 32 битных словах
6	резерв	
6	флаги	URG, ACK, PSH, RST, SYN, FIN
16	размер окна	
16	контрольная сумма	
16	указатель важной информации	ее длина в данных пакета
[32]	опции и заполнитель	
...	данные	

## Установление и разрыв соединения

Станция А пытается соединиться со станцией В

А:  $SN = ISN_A, SYN = 1 \implies B$   
 В:  $SN = ISN_B, SYN = 1, ACK = 1, data_B \implies A$   
 А:  $SN = ISN_A + 1, ACK = 1, data_A \implies B$   
 В:  $ACK N = ISN_A + length(data_A), data_B \implies A$   
 А:  $ACK N = ISN_B + length(data_B), data_A \implies B$

Станция А инициирует разрыв соединения со станцией В

А:  $FIN = 1, [data] \implies B$   
 В:  $ACK = 1, FIN = 1, [data_B] \implies A$   
 А:  $ACK = 1 \implies B$

Буферы отправки и получения

ТСР пакет делится на сегменты по размеру окна

Отправка: 

подтвержденные сегменты	можно отправлять	нельзя отправлять
-------------------------	------------------	-------------------

Получение: 

S	S	S	S		S			S	S			
ACK N	→											

Таймеры

- повторных передач
- запросов (persist timer) — 1 байт для проверки достижимости (когда надо передавать)
- контроля работоспособности (когда не надо передавать)
- времени жизни сегмента (после обмена FIN перед разрывом соединения)

## Уровень приложений

Примеры:

FTP — File Transfer Protocol — передача файлов

HTTP — HyperText Transfer Protocol —

передача файлов с поддержкой гипертекстовых ссылок

### FTP — File Transfer Protocol

Пересылка файлов и доступ к локальным файловым системам

команды управления — ТСР, порт 21 (на время сеанса)

данные — ТСР, порт 20 (для каждого файла)

взаимодействие клиент–сервер

обмениваются текстовыми командами и откликами

во время сеанса связи

### FTP, примеры некоторых команд

CWD path	сменить каталог
DELE file	удалить файл
LIST	список файлов текущего каталога
MKD name	создать каталог
MODE type	режим обмена — поток, блоки, сжатие
PASS word	помылка пароля
QUIT	выход
REIN	завершить сеанс и открыть новый
RETR file	получить файл
STOR file	отослать файл
TYPE ...	тип обмена — binary, ascii
USER [name[passwd]]	авторизация пользователя
и т.д.	

### FTP, примеры некоторых откликов

125 Data connection already open	1xx — предварительный информационный отклик
220 abc.ru FTP server ready	2xx — команда выполнена успешно
221 goodbye	
230 user Ivanov logged in	
331 password required for Ivanov	3xx — нужно уточнение (доп. параметры)
425 can't open data connection	4xx — команда не выполнялась
452 error writing file	
500 syntax error in command	5xx — ошибка в команде
501 error in parameter	
и.т.п.	

### HTTP — HyperText Transfer Protocol

TCP, порт 80

клиент–сервер, текстовые запросы и отклики

HTTP/1.1 — persistent connection

для каждого объекта запроса связь устанавливается

на некоторое время (до длительного простоя)

— with|without pipelining — запросы

последовательно | с упреждением

— параллельные соединения для разных объектов

URL — Uniform Resource Locator

<протокол>://<логин>:<пароль>@<хост>:<порт>/<URL-путь>?<параметры>#<якорь>

### HTTP, формат запроса

request line — метод URL версия CR LF

header line — название заголовка : значение CR LF

header line — ... CR LF

CR LF

entity body — данные, если надо (от формы)

```
GET /dir/file.html HTTP/1.1
Host: yandex.ru localhost:5555
Connection : close keep-alive
User-agent : Mozilla Firefox 27.0.1
Accept : text/html image/gif image/jpeg
Асcept-language : ru
<пустая строка>
данные, если надо
```

### HTTP, формат отклика

```
status line — версия код-отклика сообщение 

|    |    |
|----|----|
| CR | LF |
|----|----|


header line — название заголовка : значение 

|    |    |
|----|----|
| CR | LF |
|----|----|



|    |    |
|----|----|
| CR | LF |
|----|----|


entity body — данные (содержимое файла)
```

```
HTTP/1.1 200 OK
Connection : close
Date : Wed, 25 Mar 2014 06:32:25 GMT
Server : Apache/1.3.0 (Unix)
Last-modified : Mon, 27 Jan 2010 11:12:42 GMT
Content-length : 27631
Content-type : text/html
<пустая строка>
содержимое файла
```

### HTTP, коды отклика

1xx — information  
2xx — success  
3xx — redirection  
4xx — client error  
5xx — server error

200 OK  
301 Moved permanently

Location : новый URI  
400 Bad request  
404 Not found  
505 HTTP version not supported

### HTTP, строки заголовков

Авторизация  
сервер:  
401 Authorization required  
WWW-authenticate : детали требуемой авторизации

клиент:

Authorization : name, password

Cookies

сервер:

Set-cookie : 12345678 (имя/номер)

клиент:

Cookie : 12345678

— упрощение авторизации, статистика, индивидуальная настройка

### **НТТР, кеширование**

клиент:

If-modified-since : дата, время

— внутреннее кеширование (браузер)

— внешнее кеширование — проху server

"перехватывает" запросы клиентов

отправляет запрос от своего "имени кеширует

и рассылает по клиентам