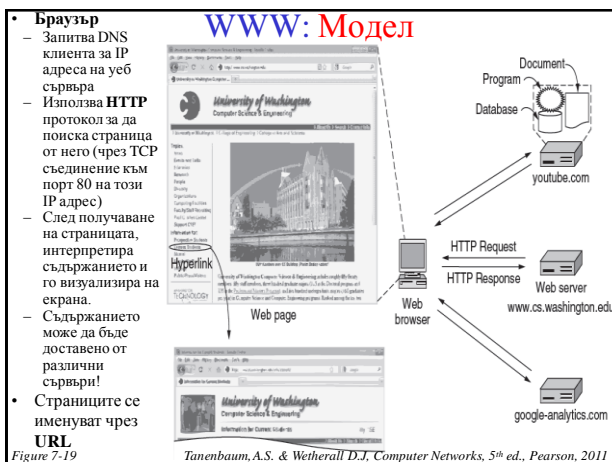


# World Wide Web (WWW)

## WWW: История

- Архитектурна рамка за достъп до взаимно свързани документи (уеб страници, съдържащи *хипертекст*) в Интернет
  - Страници, написани в HTML формат.
  - Разглеждат се с *браузър* (действа като HTML интерпретатор)
- Първоначално предложение на Тим Бърнърс-Лий, CERN (Европейски център за ядрени изследвания) - март 1989 г.
- 1. прототип (текстово-базиран) демонстриран на Hypertext'91 конференцията в Сан Антонио, Тексас, САЩ (декември 1991)
- 1. графичен браузър (Mosaic), разработен от Марк Андрийсен от Университет на Илинойс (февруари 1993 г.)
- Компанията Netscape Comm. Corp. основана от Андрийсен (1994)
- "Война на браузърите" между Netscape Navigator и MS Internet Explorer (1995-1998)
- World Wide Web Consortium (W3C) създаден през 1994 г. ([www.w3.org](http://www.w3.org))

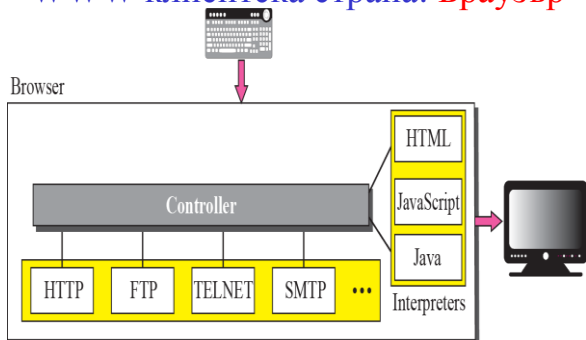


## URL (Uniform Resource Locator)

Protocol	Host	Port	Path
Name	Used for	Example	
http	Hypertext (HTML)	http://www.ee.uwa.edu/~rob/	
https	Hypertext with security	https://www.bank.com/accounts/	
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README	
file	Local file	file:///usr/suzanne/prog.c	
mailto	Sending email	mailto:JohnUser@acm.org	
rtsp	Streaming media	rtsp://youtube.com/montypython.mpg	
sip	Multimedia calls	sip:eve@adversary.com	
about	Browser information	about:plugins	

- Нова RFC 2141 схема **URN** (Universal Resource Name) – подмножество на **URI** (Uniform Resource Identifier):
  - Не указва към конкретен хост
  - Елиминира проблема на прекалено използваните страници
  - Намалжава мрежовия трафик като позволява достъп до няколко отдалечени копия

## WWW клиентска страна: Браузър



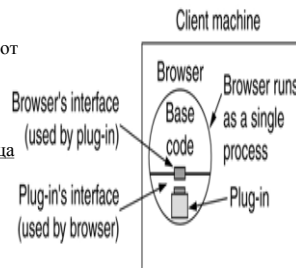
Браузърът може да използва различни протоколи за различни видове ресурси

## WWW клиентска страна: Plug-in

- Уеб страниците могат да съдържат не само HTML, но също така PDF, GIF, JPEG, MP3 и т.н.
- Проблем когато браузърът се натъкне на страница, която не може да интерпретира.
- 2 решения за предотвратяване усложняването на браузъра

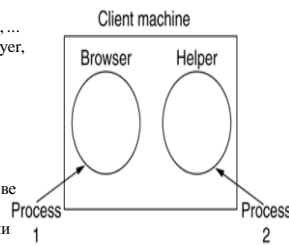
### Plug-in / приставка:

- Софтуерен компонент, изтеглен от Интернет и инсталиран като разширение на браузъра.
- Изпълнява се вътре в него
- Има достъп до текущата страница и може да промени визуализацията ѝ
- Трябва да реализира набор от процедури, специфични за браузъра, така че той да може да ги извиква.



## WWW клиентска страна: Приложение-помощник (*helper application*)

- Независима програма, работеща като отделен процес (*external viewer*).
  - Например, Adobe Acrobat, MS Word, ...
  - Или Real Player, Windows Media Player, Winamp за аудио, и т.н.
- Не използва услугите на браузъра
- Стартира се автоматично с указване на файла, съдържащ съдържанието за визуализиране.
- Може да бъде асоциирана с файловото разширение за да отваря локални файлове
- Използва MIME типове
  - Например, `application/pdf` за PDF или `application/vnd.ms-word` за MS Word, ...
  - Например, `image/x-photoshop` за Adobe Photoshop, ...
- Потенциален конфликт, когато са налични няколко помощника за един и същ подтип, например за `video/mpg`, `audio/mp3`, ...



## WWW сървърна страна



- Многонишков веб сървър с *front-end* и *k* обработващи модула
- Всеки модул извършва серия от стъпки:
  - Проверява името на заявената веб страница
  - Извършва контрол на достъпа до веб страницата (например, ограничено ползване само за потребители вътре във фирмата или IP пространството)
  - Извлича исканата страница от кеша или диска или стартира програма за създаването ѝ
  - Определя останалата част от отговора (например, MIME тип за изпращане)
  - Отговаря на клиента (използва 1 TCP съединение за извличане на няколко страници)
  - Прави запис в регистрационния файл (*log file*) на сървъра

Figure 7-21 Tanenbaum, A.S. & Wetherall D.J., Computer Networks, 5th ed., Pearson, 2011

## Web Statelessness and Cookies

- Не се запазва информация за състоянието (*stateless*)
- Без концепция за потребителска сесия (*login session*)
  - Браузърът изпраща заявка към сървъра и получава обратно файл
  - Сървърът забравя за клиента
- Проблеми за сървъра:
  - Как да прави разлика между заявки на регистрирани и други потребители?
  - Как да следи съдържанието на кошиците за пазаруване (при електронна търговия)?
- IP адресът не върши работа, ако се използва NAT!
- Решение: използване на **cookies / курабийки**
  - Малки файлове/strings (< 4kB), съдържащи до 5 полета
  - Създавани от сървъра и съхранявани на твърдия диск на клиента.
- Противоречиво използване
  - Могат да следят *on-line* поведението на потребителя (*spyware*)
  - Могат да съдържат вируси
- Потребителски контрол – т.е. кои да се приемат/отхвърлят

Domain	Path	Content	Expires	Secure
toms-casino.com	/	CustomerID=497793521	15-10-02 17:00	Yes
joes-store.com	/	Cart=1-00501;1-07031;2-13721	11-10-02 14:22	No
aportal.com	/	Prefs=Stk:SUNW+ORCL;Spt:Jets	31-12-10 23:59	No
sneaky.com	/	UserID=3627239101	31-12-12 23:59	No

## HTTP (HyperText Transfer Protocol)

- Специфицира съобщенията, разменяни между браузъри и веб сървъри (RFC 2616).
- Всяко взаимодействие се състои от ASCII заявка, последвано от MIME-подобен отговор.
- Браузърът се свързва със сървъра по TCP съединение към порт 80
- HTTP 1.0
  - Изпраща се 1 **заявка** и се получава 1 **отговор**; TCP съединението се разпада.
  - Значителни допълнителни разходи, тъй като няколко TCP съединения са необходими за 1 веб страница, съдържаща голям брой икони, изображения и др.
- HTTP 1.1
  - Постоянни съединения (*persistent connections*)
  - Множество заявки/отговори могат да се обменят по 1 TCP съединение
  - Могат да се използват **поточни заявки** (*pipeline requests*), т.е. да се изпрати **заявка 2** преди получаването на **отговор 1**.
- HTTP 2 (2015 г.)
  - Компресия на HTTP заглав. част
  - Сървърът може да използва *push*

## HTTP: Съединения

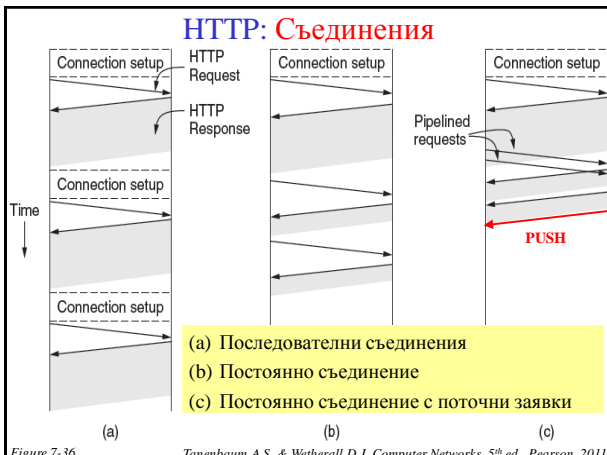


Figure 7-36

Tanenbaum, A.S. & Wetherall D.J., Computer Networks, 5th ed., Pearson, 2011

## HTTP: Опростено функциониране

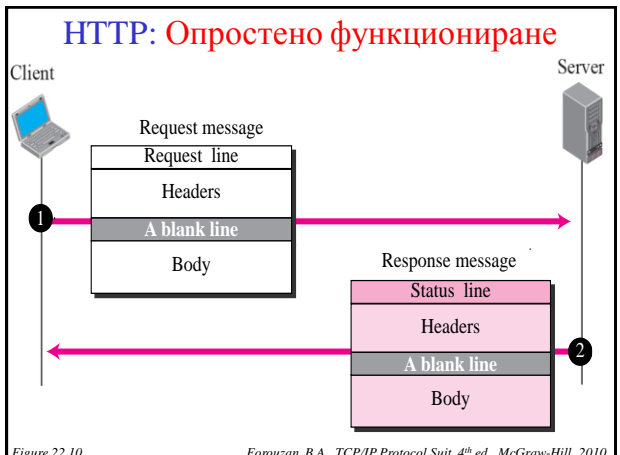


Figure 22.10

Forouzan, B.A., TCP/IP Protocol Suite, 4th ed., McGraw-Hill, 2010

**HTTP: Заявки**

- 1 или повече ASCII текстови редове
  - 1. ред: request line (method)
  - Допълнителни редове: заглавни части на заявката (headers)

Method	Description
GET	Read a Web page
HEAD	Read a Web page's header
POST	Append to a Web page
PUT	Store a Web page
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Connect through a proxy
OPTIONS	Query options for a page

- Допълнителни специфични за обекта (невградени) методи също могат да се използват

Figure 7-37

Tanenbaum, A.S. &amp; Wetherall D.J., Computer Networks, 5th ed., Pearson, 2011

**HTTP: Отговори**

- 1. ред: status line
  - 3-цифрен код указващ дали заявката е удовлетворена или не (и защо)

Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later

- Допълнителни редове: заглавни части на отговора (headers)

Figure 7-38

Tanenbaum, A.S. &amp; Wetherall D.J., Computer Networks, 5th ed., Pearson, 2011

**HTTP: Заглавни части на съобщенията**

Header	Type	Contents
User-Agent	Request	Information about the browser and its platform
Accept	Request	The type of pages the client can handle
Accept-Charset	Request	The character sets that are acceptable to the client
Accept-Encoding	Request	The page encodings the client can handle
Accept-Language	Request	The natural languages the client can handle
If-Modified-Since	Request	Time and date to check freshness
If-None-Match	Request	Previously sent tags to check freshness
Host	Request	The server's DNS name
Authorization	Request	A list of the client's credentials
Referer	Request	The previous URL from which the request came
Cookie	Request	Previously set cookie sent back to the server
Set-Cookie	Response	Cookie for the client to store
Server	Response	Information about the server

Figure 7-39

Tanenbaum, A.S. &amp; Wetherall D.J., Computer Networks, 5th ed., Pearson, 2011

**HTTP: Заглавни части на съобщенията (прод.)**

Content-Encoding	Response	How the content is encoded (e.g., gzip)
Content-Language	Response	The natural language used in the page
Content-Length	Response	The page's length in bytes
Content-Type	Response	The page's MIME type
Content-Range	Response	Identifies a portion of the page's content
Last-Modified	Response	Time and date the page was last changed
Expires	Response	Time and date when the page stops being valid
Location	Response	Tells the client where to send its request
Accept-Ranges	Response	Indicates the server will accept byte range requests
Date	Both	Date and time the message was sent
Range	Both	Identifies a portion of a page
Cache-Control	Both	Directives for how to treat caches
ETag	Both	Tag for the contents of the page
Upgrade	Both	The protocol the sender wants to switch to

Figure 7-39

Tanenbaum, A.S. &amp; Wetherall D.J., Computer Networks, 5th ed., Pearson, 2011

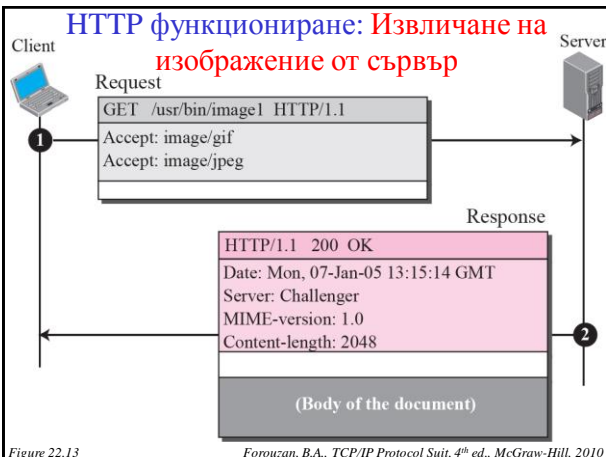
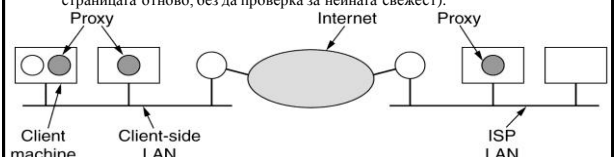
**HTTP функциониране: Извличане на изображение от сървър**

Figure 22.13

Forouzan, B.A., TCP/IP Protocol Suit, 4th ed., McGraw-Hill, 2010

**WWW: Подобрения на производителността**

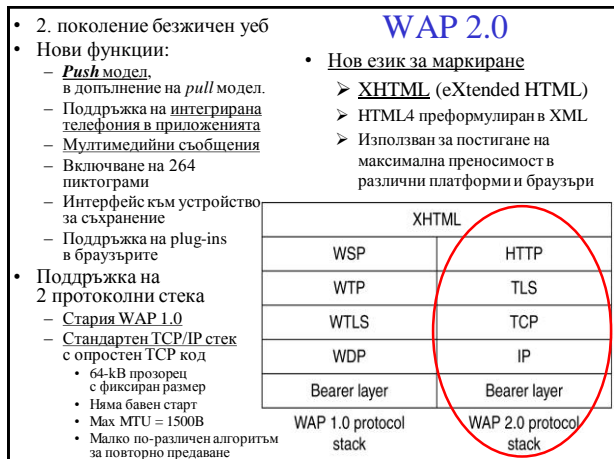
- Сървъри – репликиране (replication)**
  - Отражения (mirroring) – възпроизвеждане съдържанието на един сървър на множество други разпръснати сървъри
  - Сървърни ферми (server farms)
- Клиенти – кеширане (caching)**
  - Браузърите кешират отговорите с цел използване при бъдещи подобни заявки
  - Съкращаване времето за отговор и намаляване натоварването на мрежата
  - Използват полета в заглав. част + правила за да определят дали кешираното копие е свежо
- Проксита – споделяне на кеш сред множество потребители**
  - Иерархично кеширане (последователно изпробване на няколко кеша)
  - Изключения: динамични страници (чиито параметри могат да бъдат различни следващия път) и страници, които изискват разрешение (authorization) (сървърите могат да инструктират всички проксита по пътя да не използват страницата отново, без да проверка за нейната свежест).





## Мобилен уеб (Mobile Web)

- Трудности за мобилни телефони, сърфиращи в уеб пространството:
  - Сравнително малките екрани затрудняват визуализирането на големи страници и изображения
  - Ограничените входни възможности правят досадно въвеждането на URL адреси или дълги вход. записи
  - Пропускателната способност на мрежата е ограничена
  - Свързаността може да е с прекъсвания
  - Изчислителната мощност е ограничена
- Ранен подход – разработване на **нов протоколен стек**, пригоден към мобилни устройства с ограничени възможности.
  - WAP (Wireless Application Protocol)
  - Оптимизиран за **бавни комуникационни линии** и **нисък клас мобилни устройства** (със слаб процесор, малко памет, малък екран, ...)



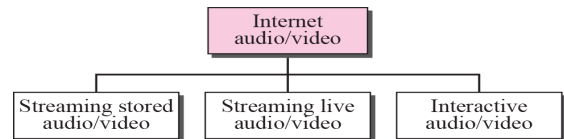
## Мобилен уеб (прод.)

- Значително разрастване на пропускателната способност на мрежите и възможностите на мобилните устройства
- Днешните мобилни телефони могат да използват (прости) уеб браузъри
- Уеб сървърите могат да установят дали да отговорят с мобилна или пълна версия на дадена уеб страница чрез разглеждане на заглав. части на заявката
  - User-agent header** - идентифицира типа на браузъра
    - Ако е мобилен, сървърът връща мобилна версия на страницата, например с малки изображения, по-малко текст, проста навигация...
  - Големите уеб сайтове създават мобилни уеб версии на тяхното съдържание
    - Например, чрез използване на компресия за намаляване размера на страниците при предаване (защото комуникационните разходи > изчислителните разходи)
- Транскодиране/Transcoding** (трансформация на съдържание)
  - Допълващ подход – за страници, непредназначени за мобилни устройства.
  - Изпълняван от **междинен** компютър
    - Получава заявка от мобилно устройство
    - Изтегля търсеното съдържание от сървъра
    - Трансформира го в мобилно съдържание, което най-добре съответства на мобил. устройство.
    - Например, чрез преформатиране на изображенията за да се намали резолюцията.
- Използване на HTTP, TCP и IP, но с допълн. **компресия на заглав. части**
  - ROHC** (RObust Header Compression)
  - Намаляване на допълн. разходи (*overhead*)
  - Особено при използване на бавни комуникационни линии

# Мултимедия

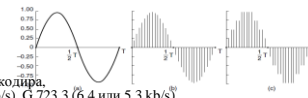
## Мултимедия

- Комбинация от 2 или повече медии
  - Възпроизведени по време на определен времеви интервал
  - Обикновено с някакво взаимодействие с потребителя
- **Аудио** (компресирано)
  - Поточно съхранено аудио (*аудио при поискване*)
  - Поточно на живо (broadcast) аудио (например, *Интернет радио*)
  - Интерактивно аудио (*Интернет телефония - VoIP*)
- **Видео** (компресирано)
  - Поточно съхранено видео (*видео при поискване*)
  - Поточно на живо (broadcast) видео (например, *Интернет телевизия - IPTV*)
  - Интерактивно видео (*видео конферентна връзка*)

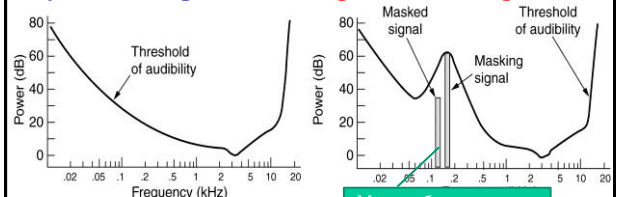


## Цифрово аудио

- Аудио-вълната е (едномерна) акустична вълна
- Възприемана като звук от слушател при преминаване през ухото му (тъпанчето вибрира, причинявайки вибрации в малките кости на вътрешното ухо, които изпращат нервни импулси към мозъка)
- Ухото е чувствително към звукови вариации с времетраене няколко *ms*!
  - За разлика от него окото не забелязва промени в светлинното ниво с времетраене едва няколко *ms*
  - **Резултат:** вариациите на закъснението ( *jitter* ) се отразяват повече на качеството на звука, отколкото на качеството на изображението!
- Аналогов аудио-сигнал се **цифровизира** (според теоремата Найкуист/Котелников) и се превръща в цифров сигнал
- **Аудио-компресия** при предаване:
  - 64-kb/s цифровизирана реч
  - **Предсказуемо кодиране (predictable coding)** – само разликата между пробите се кодира, например, **GSM** (13 kb/s), **G.729** (8 kb/s), **G.723.3** (6.4 или 5.3 kb/s)
  - 1.411-Mb/s CD-качество стерео музика (2 x 44100 проби/сек от 16 бита всяка)
  - **Вълново кодиране (waveform coding)** – амплитудата на всяка честотна компонента се кодира по минимален начин, използвайки по възможност най-малко битовете.
  - **Кодиране по възприятие (perceptual coding)** – експлоатира някои недостатъци в слуховата система на човека; някои звуци маскират други звуци (напр. в **MP3**).



## Аудио компресия: Кодиране по възприятие



Въз основа на 2 вида на маскиране:

- **Честотно маскиране**
  - Силен звук в една честотна лента покрива слаб звук в друга лента
  - Например, трудно е да се чуе какво казва партньорът по танци при силна музика.
  - **Придаване:** констатираните маскирани честоти могат да се изпуснат при аудио кодирането, като по този начин се спестяват битовете!
- **Времево маскиране**
  - По-слабият звук не може да се улови от човешкото ухо за известно време, дори след прекратяване на по-силния звук
  - На ухото му трябва време за адаптация!
  - **Придаване:** дори след спиране на мощния сигнал, познания за свойствата на времето му маскиране позволяват да продължи пропускането на маскираните честоти (при аудио кодирането) за някакъв интервал от време!

Figure 7-43

Tanenbaum, A.S. & Wetherall D.J., Computer Networks, 5th ed., Pearson, 2011

Може да бъде пропуснат при аудио кодирането

## Цифрово видео

- Последователност от кадри, всеки представляващ правоъгълна решетка от елементи (**пиксели**).
- По няколко бита на пиксел
  - Например, 8 бита за представяне на 256 нюанса на сивия цвят.
  - Например, 8 бита за всеки RGB цвят, което води до повече от 16 милиона цвята (човешкото око дори не може да разграничи съседните).
- Възпроизвеждани най-малко 25 кадъра/сек
  - РС мониторите сканират изображенията, съхранени в паметта, с по-висока честота за да се премахне трептенето (*flickering*).
  - Например, при 75Hz монитор визуализира един и същ кадър 3 пъти поред.
- РС мониторът трябва да се захранва със скорост > 472 Mb/s
  - Например, при 24 бита/пиксел, 25 кадъра/сек, резолюция 1024 x 768.
- Затова се налага масивна **видео компресия!**

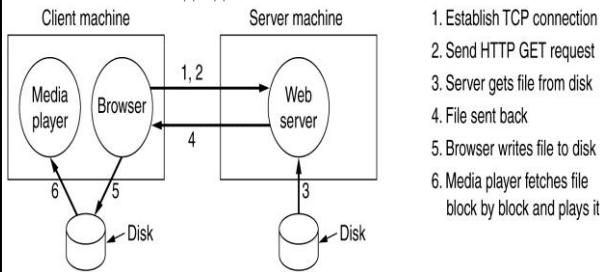
## Видео компресия

- **Компресиране със загуба** на информация в източника (**кодиране**) и декомпресиране при получаване (**декодиране**)
  - Видеоот след декодиране може да бъде малко **по-различно** от оригиналното видео
  - Малка загуба на информация може да доведе до голяма степен на компресия!
- За видео, предавано **не в реално време:**
  - **Асиметрия**
    - Например, филм кодиран само веднъж и съхраняван на мултимедийен сървър, но декодиран много пъти при заявка от различни клиенти.
  - Кодиращ алгоритъм
    - Бавен, сложен и изискващ скъп хардуер.
  - Декодиращ алгоритъм
    - Бърз, прост и неизискващ скъп хардуер.
- За видео, предавано **в реално време:**
  - Кодиране по време на предаване (*on-the-fly*)
  - Бавно кодиране е неприемливо!
  - Различни алгоритми и/или параметри се използват, често с по-малка степен на компресия.



### Поточно аудио/видео (съхранено): Чрез използване на уеб сървър

- Слушане на аудио през Интернет (*музика при поискване*)
- Гледане на видео през Интернет (*видео при поискване*)
- Обикновен подход:



- **Недостатък:** целият аудио/видео файл трябва да бъде изтеглен, преди започване на възпроизвеждането му.

### Поточно аудио/видео (съхранено): Чрез използване на метафайл

- Подобен подход
- **Метафайл**
  - Много малък файл (просто указващ към музиката/видеото)
  - Съдържащ URL
  - Изтеглен от браузъра в у/диск
- Браузърът извиква мултимедиян плейър като помощник (*helper*) и му предава името на метафайла
- Мултимедияният плейър се обръща към отдалечен (уеб или медиян) сървър за дад. песен и/или видео
- Браузърът не участва повече

Figure 25.12 Forouzan, B.A., TCP/IP Protocol Suite, 4th ed., McGraw-Hill, 2010

### Мултимедиян плейър: Задачи

- **Управление на графичен потребителски интерфейс (GUI)**
  - Например, симулиращ стерео плейър с бутони, копчета, плъзгачи, визуални дисплеи, сменяеми предни панели (*skins*) и т.н.
- **Борба с грешки при предаване** (главно загуба на пакети)
  - Промяна на реда на предаване на пробите
    - Например, когато пакетите пренасят алтернативни проби, загубата на пакет намалява временно резолюцията, а не създава временен престой при възпроизвеждане.
    - FEC (Forward Error Correction)
      - Например, контрол по четност, обхващащ няколко пакета.
- **Декомпресия**
  - Интензивна (от гледна точка на използване на изчислител. ресурси), но иначе лесна.
  - Трябва да позволява декомпресиране въпреки загуба на пакети (за UDP-базиран транспорт)
    - Ако по-късни данни са компресирани относително спрямо по-ранни данни, те не могат да бъдат декомпресирани, докато по-ранните данни не се декомпресират.
    - Ето защо MPEG-1 използва кадри, които могат да бъдат декодирани независимо от други кадри за да се възстанови възпроизвеждането при загуба на предишни кадри.
- **Борба с вариациите на закъснението (jitter)**
  - Чрез буферизиране преди започване на възпроизвеждането
  - Плейърът буферизира входа от медияния сървър (за 10-15 сек) и възпроизвежда от буфера, а не директно от мрежата.

### Борба с грешки при предаване:

#### Промяна на реда на предаване на пробите

- ‘Умно’ кодиране конвертира загубата на пакети в по-ниско качество на възпроизвеждането, а не във временен престой.
- Например, за **некомпресирани аудио проби** (RFC 3119):
  - Пакетът може да съдържа 220 стерео проби, всяка съдържаща двойка 16-битови числа (≈ 5 msec музика)
  - Всички нечетни проби (в 10-msec интервал) се изпращат в един пакет
  - Всички четни проби се изпращат в следващия пакет
  - Загубен пакет не представлява 5-msec празнина в музиката, а просто загуба на всяка втора проба в 10-msec интервал.
  - Тази загуба може да бъде компенсирана чрез **интерполация** на предишни и следващи проби



Figure 7-25 Tanenbaum, A.S. & Wetherall D.J., Computer Networks, 5th ed., Pearson, 2011

### Борба с грешки при предаване: FEC

- Например, чрез изпращане на 5. контролен пакет P за всеки 4 пакета с данни (A, B, C, D).
- P съдържа контролни битове, които са XOR суми на съответните битове, съдържащи се в пакетите с данни.
- Ако 1 пакет с данни (например B) се изгуби (или закъснее много), неговите битове могат да бъдат възстановени по следния начин:  
$$B = P \oplus A \oplus C \oplus D$$
- Ако 2 пакета в група са изгубят, възстановяване на данните не е възможно.
- **Недостатък:**
  - Излишъкът е 20%
  - Така се изискванията към честот. лента са с 20% по-високи
  - Повишена латентност на декодирането (изчакване пристигането на контролния пакет за да може да започне възстановяването на липсващия пакет в групата)

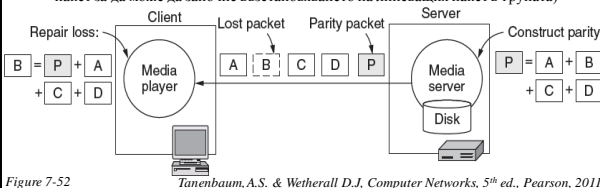
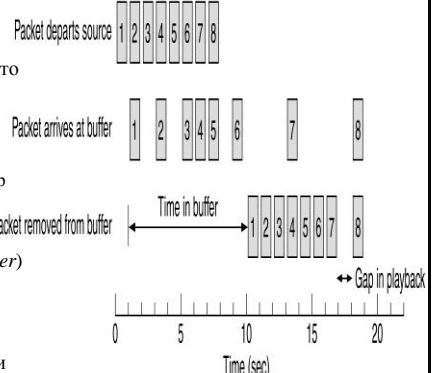


Figure 7-52 Tanenbaum, A.S. & Wetherall D.J., Computer Networks, 5th ed., Pearson, 2011

### Борба с вариациите на закъснението: Буферизиране

- Потокът се буферизира от плейъра преди започване на възпроизвеждането
  - Например, комерсиалните уеб сайтове за **поточно аудио/видео** използват буфер от 10-15s
- Изглажда вариациите на закъснението (*jitter*)
  - Важно за аудио/видео по поръчка/заявка
  - Може да изисква по-големи буфери



## Борба с вариациите на закъснението: Буфериране (прод.)

2 подхода за поддържане на буфера пълен:

- **Pull сървър**
  - Докато има свободно място в буфера, мултимедийният плейър продължава да изпраща заявки към сървъра за допълнителни блокове.
  - Рядко използван
  - **Недостатък:** Излишно изпращане на заявки за данни
- **Push сървър**
  - Сървърът изпраща данните към плейъра
  - 2 възможности:
    - 1) Сървърът работи с нормална скорост на възпроизвеждане
      - ✓ Простота; не са необходими управляващи съобщения (и в двете посоки).
      - ✓ Честотната лента може да не е достатъчна: обикновено видео клип се кодира в различни резолюции с цел да се позволи на потребителя да избере резолюция, подходяща за скоростта на неговата Интернет връзка.
      - ✓ Вариациите на закъснението са проблематични!
    - 2) Сървърът работи по-бързо (вж. следващия слайд)

- **Предимство:** сървърът може да **навакса**, ако изостане.
- **Недостатък:** възможно **препълване на буфера** на плейъра, ако сървърът предава данни по-бързо, отколкото се консумират.
- **Решение:** плейърът определя 2 прага (*water marks*)

- Горен праг
  - При достигането му, плейърът съобщава на сървъра да спре (**PAUSE**).
  - Междувременно данните продължават да пристигат, докато сървърът възприеме командата за пауза, но не трябва да достигат макс. капацитет на буфера.
- Долен праг
  - При достигането му, плейърът съобщава на сървъра да поднови предаването (**PLAY**).
  - Трябва да бъде така избран, че буферът да не се изпразни.

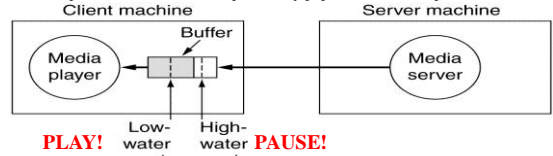


Figure 7-54 Tanenbaum, A.S. & Wetherall D.J., Computer Networks, 5th ed., Pearson, 2011

## Real Time Streaming Protocol (RTSP)

- RFC 2326
- Позволява на мултимедийен плейър да контролира дистанционно **push сървър** чрез следните команди:

Command	Server action
DESCRIBE	List media parameters
SETUP	Establish a logical channel between the player and the server
PLAY	Start sending data to the client
RECORD	Start accepting data from the client
PAUSE	Temporarily stop sending data
TEARDOWN	Release the logical channel

Figure 7-55 Tanenbaum, A.S. & Wetherall D.J., Computer Networks, 5th ed., Pearson, 2011

## Предаване на поточно (съхранено) аудио/видео: Чрез RTSP

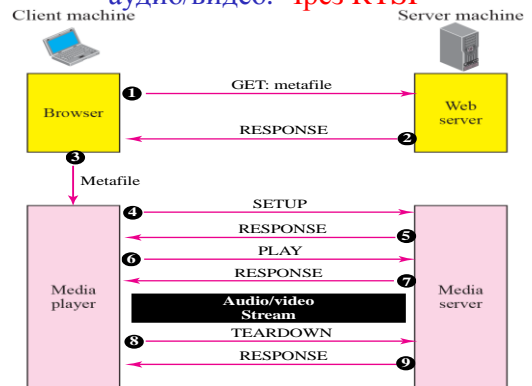


Figure 25.13 Forouzan, B.A., TCP/IP Protocol Suite, 4th ed., McGraw-Hill, 2010

- Излъчване "на живо" на аудио/видео през Интернет
  - Например, комерсиални радио/TV станции, излъчващи своите програми "на живо" едновременно в ефир и по Интернет.
  - Интернет радио
  - Интернет телевизия (IPTV, IP TeleVision)
- Буфериране в клиентската страна все още е необходимо за изглаждане на вариациите на закъснението
  - Освен това може да се използва за предоставяне на допълнителни функции като пауза и/или превъртане назад
  - Необходими са по-големи буфери (10-15 сек. забавяне при стартиране)
- Разлики с поточно (съхранено) аудио/видео:
  - Винаги се излъчва със същата скорост, с която се генерира и възпроизвежда.
  - От една точка към много други точки (множество слушатели)
- По-ефективен начин за работа е **multicasting over RTP/RTSP**
- На практика обаче, TCP може да се ползва за преминаване през защитни стени (*firewalls*).
- Медийни формати: RealAudio, Windows Media Audio, MP3, Vorbis (подобен на MP3, но с *open-source*).

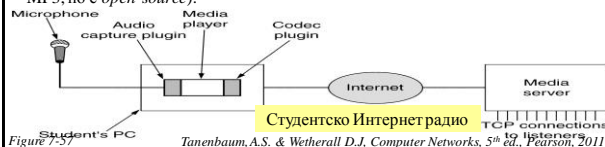


Figure 7-57 Tanenbaum, A.S. & Wetherall D.J., Computer Networks, 5th ed., Pearson, 2011

## Multicast поточно аудио/видео: С използване на контролен пакет за проверка по четност

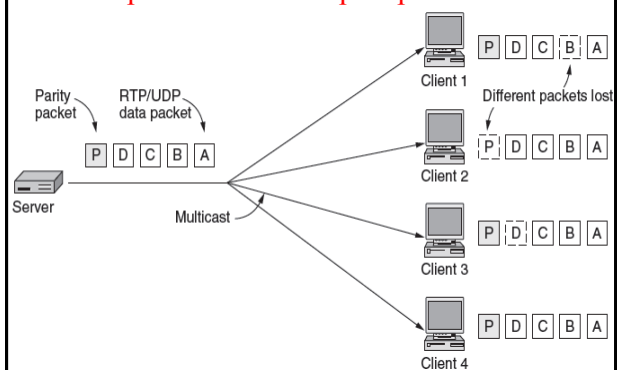


Figure 7-56 Tanenbaum, A.S. & Wetherall D.J., Computer Networks, 5th ed., Pearson, 2011

## Конференции в реално време

2 подхода:

- ITU препоръка **H.323** (преработена 1998)
  - Типичен **тромав** стандарт на телефонната индустрия
  - Специфицира пълен протоколен стек, определящ точно какво е позволено и какво е забранено.
  - Предимство:** много добре дефинирани протоколи във всеки слой
  - Недостатък:** голям, сложен, негъвкав, трудно адаптиращ се към новите приложения.
- SIP** протокол (на IETF)
  - Типичен Интернет протокол
  - Оперира чрез обмен на кратки редове ASCII текст
  - Лек/несложен**
  - Работи добре с други Интернет протоколи, но по-малко добре с протоколи за сигнализация в телефонни системи.
  - Модулен и гъвкав** – може да се адаптира към бъдещи приложения
- И двата подхода:
  - Позволяват повиквания между два и повече участника
  - Използват компютри и/или телефони като крайни възли
  - Поддържат договаряне на параметри, шифриране, RTP/RTCP.

## H.323 v. SIP

Item	H.323	SIP
Designed by	ITU	IETF
Compatibility with PSTN	Yes	Largely
Compatibility with Internet	Yes, over time	Yes
Architecture	Monolithic	Modular
Completeness	Full protocol stack	SIP just handles setup
Parameter negotiation	Yes	Yes
Call signaling	Q.931 over TCP	SIP over TCP or UDP
Message format	Binary	ASCII
Media transport	RTP/RTCP	RTP/RTCP
Multiparty calls	Yes	Yes
Multimedia conferences	Yes	No
Addressing	URL or phone number	URL
Call termination	Explicit or TCP release	Explicit or timeout
Instant messaging	No	Yes
Encryption	Yes	Yes
Status	Widespread, esp. video	Alternative, esp. voice

Table 7-63

Tanenbaum, A.S. & Wetherall D.J., Computer Networks, 5th ed., Pearson, 2011

## H.323: Архитектурен модел

- Controls LAN PCs under its jurisdiction (*zone*)
- Grants BW to PCs on their request
- Avoids oversubscribing the outgoing line to support needed QoS

Speaks both Internet and PSTN protocols

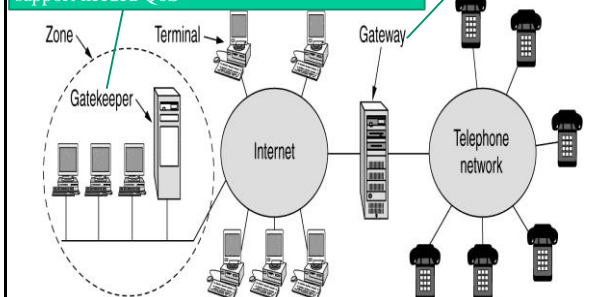


Figure 7-58

Tanenbaum, A.S. & Wetherall D.J., Computer Networks, 5th ed., Pearson, 2011

## H.323: Протоколен стек

- E.g. **G.711** (64 kbps uncompressed speech)
- E.g. **G.723.3** (6.4/5.3 kbps (compressed speech))

PC-to-gatekeeper exchange of Registration/Admission/Status (RAS) info – join/leave zone, request/return bandwidth, status updates etc.

Establishing /releasing connections, providing dial tones, making ringing sounds, etc.

Call parameters negotiation: codec, compression algorithm, bit rate, video/ conference handling.

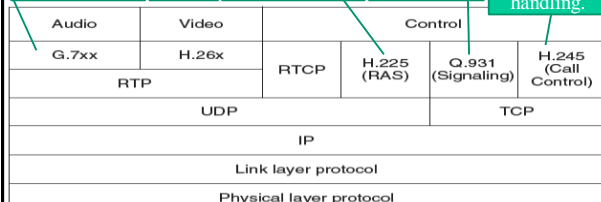


Figure 7-59

Tanenbaum, A.S. & Wetherall D.J., Computer Networks, 5th ed., Pearson, 2011

## H.323: Логически канали

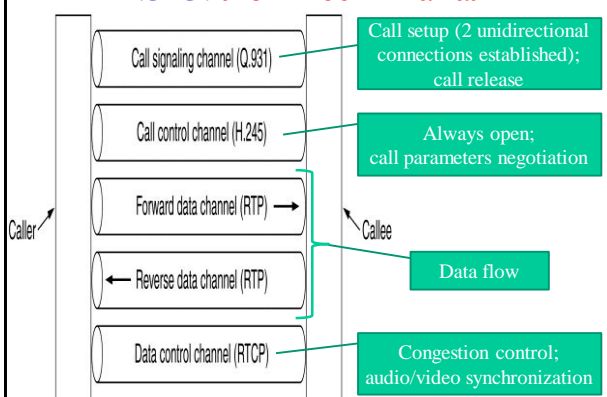


Figure 7-60

Tanenbaum, A.S. & Wetherall D.J., Computer Networks, 5th ed., Pearson, 2011

## H.323: Функциониране

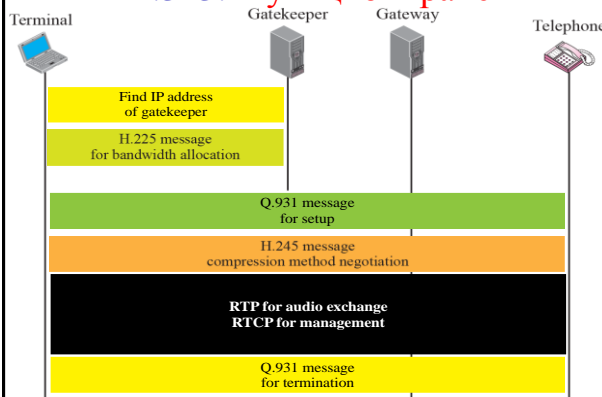


Figure 25.27

Forouzan, B.A., TCP/IP Protocol Suite, 4th ed., McGraw-Hill, 2010



# Доставка на съдържание

## Доставка на съдържание

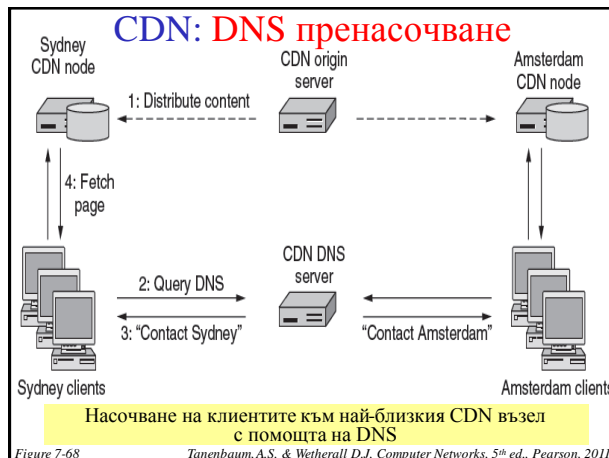
- Нова парадигма с голямо приложение напоследък
  - Например, доставка на съхранени видеоклипове, филми и др.
- Различна задача – различни изисквания към мрежата
  - Мястото, откъдето се доставя, няма значение.
    - Например, за потребителя няма голямо значение откъде точно се доставя поточното видео съдържание, стига да е от компютър, който може да осигури най-добрата (бърза) услуга.
  - Някои сайтове за съдържание са много по-популярни от други
    - Например, YouTube генерира 10% от Интернет трафика днес.
  - Големите доставчици на съдържание изграждат свои собствени мрежи за доставка на съдържание (*content delivery networks, CDNs*), които използват центрове за данни, разпръснати по света за по-добра производителност и достъпност.
  - Алтернатива са P2P (peer-to-peer) мрежите – колекция от компютри, споделящи съдържание помежду си, без специално осигурени сървъри или централен контролен пункт.

## Мрежи за доставка на съдържание (CDNs)

- Доставчик на съдържание
  - Разполага едно и също съдържание на различни места
  - Насочва клиента към 'най-близкото' място
- Дървовидна структура
  - Съдържанието се разпределя до колкото се може повече клиенти
  - По-кратко RTT
  - Общото натоварване се минимизира
- Основен въпрос: Как клиентите да ползват това дърво?
  - Прокси сървъри
  - Огледални сървъри
  - DNS пренасочване

Figure 7-67

Tanenbaum, A.S. &amp; Wetherall D.J., Computer Networks, 5th ed., Pearson, 2011



## Peer-to-Peer (P2P) мрежи

- Мрежи за споделяне на файлове, създадени от компютри на потребителите, които обединяват своите ресурси за формирането на система за доставка на съдържание.
  - Компютрите са равнопоставени (*peers*), защото всеки един едновременно действа и като *клиент* (извличайки съдържание от други компютри), и като *сървър* (предоставяйки свое съдържание).
- Файловете са разделени на части (chunks)
  - Всеки потребител може едновременно да получава нови части на даден файл отнякъде и същевременно да изпраща другите получени преди това части!
- Нямат специална инфраструктура (за разлика от CDN)
- Самостоятелно мащабиране
- BitTorrent** е сред най-популярните P2P протоколи (с отворен стандарт)
- Проблеми за решаване при BitTorrent споделяне:
  - Как да се откриват равнопоставените?
  - Как да се размножава съдържанието между равнопоставените с цел по-скорошно изтегляне?
  - Как равнопоставените да се насърчават взаимно за да предоставят съдържание помежду си?

