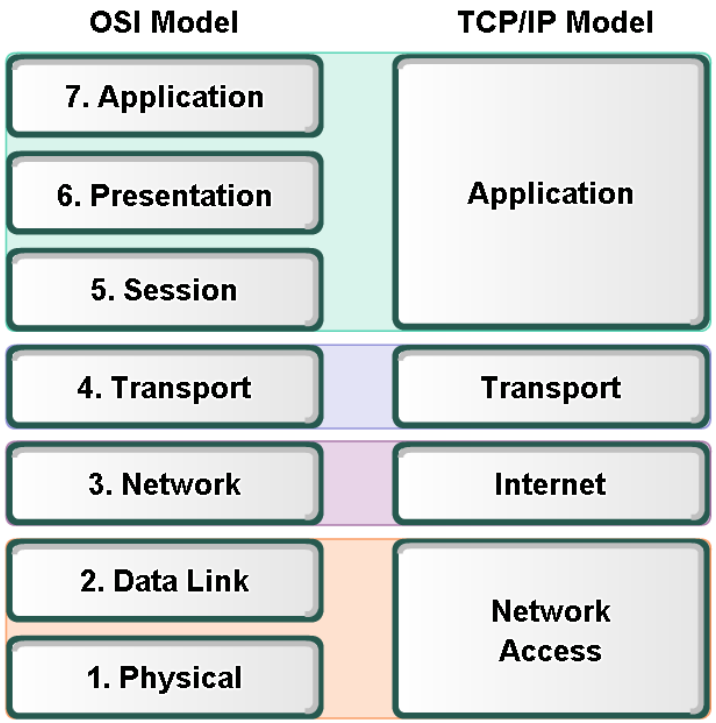
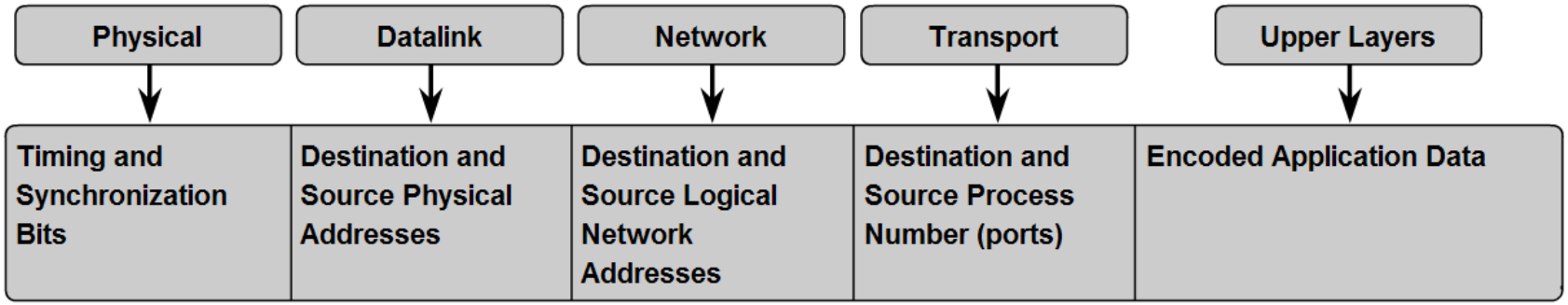
Приложен слой

проф. д-р инж. Венета Алексиева

Приложен слой

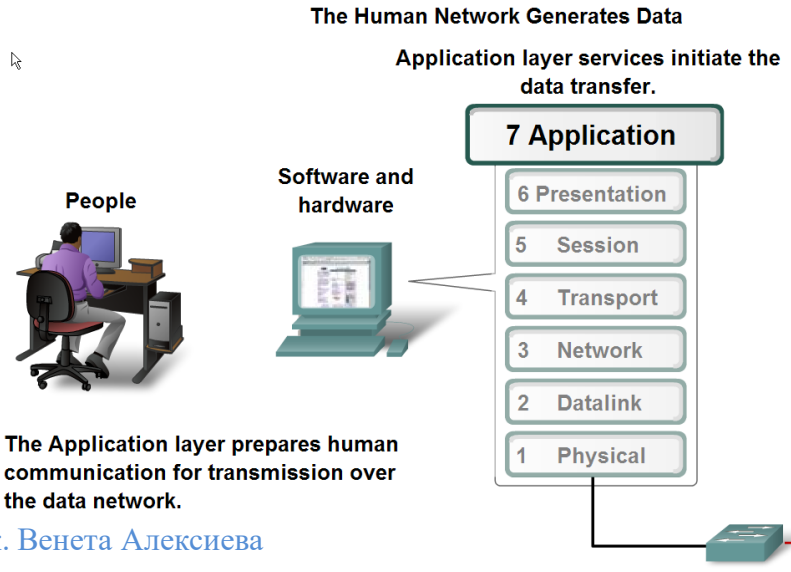


The key parallels are in the Transport and Network layers.



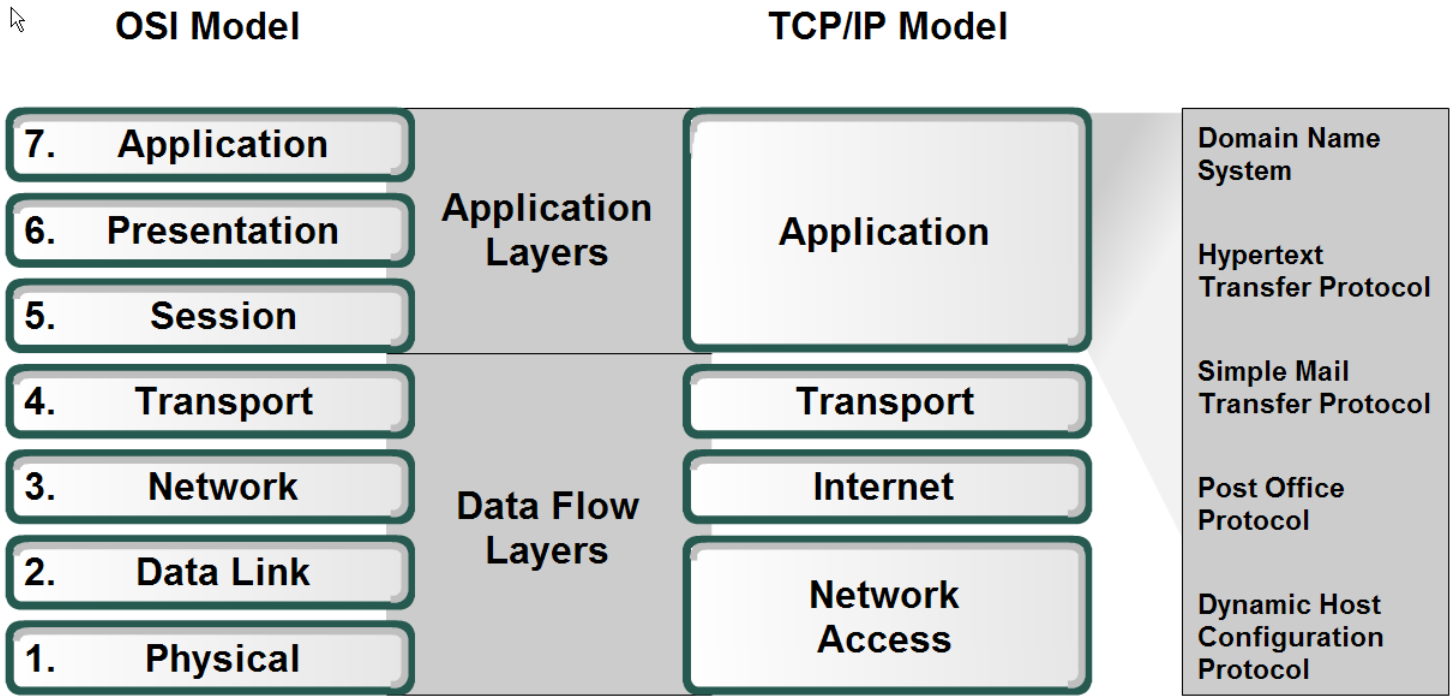
Приложен слой

- Предоставя мрежови услуги на приложенията
- Подготвя потребителските данни за предаване/получаване към/от мрежата
- Занимава се с действията по идентифициране на страните по комуникацията, качеството на услугите, идентифициране на потребителите и дефиниране синтаксиса на данните
- Използва специфични приложни протоколи

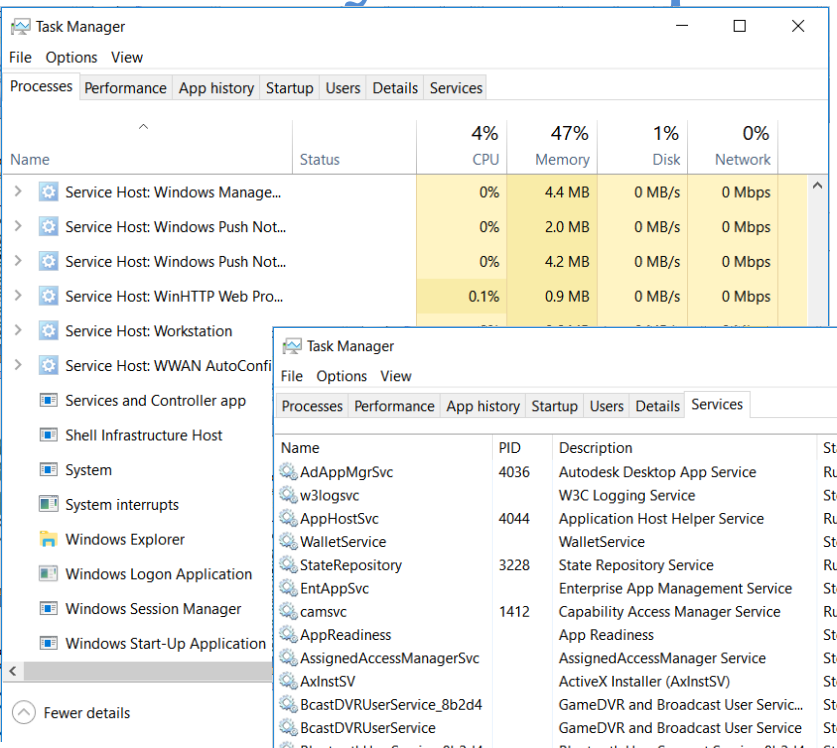


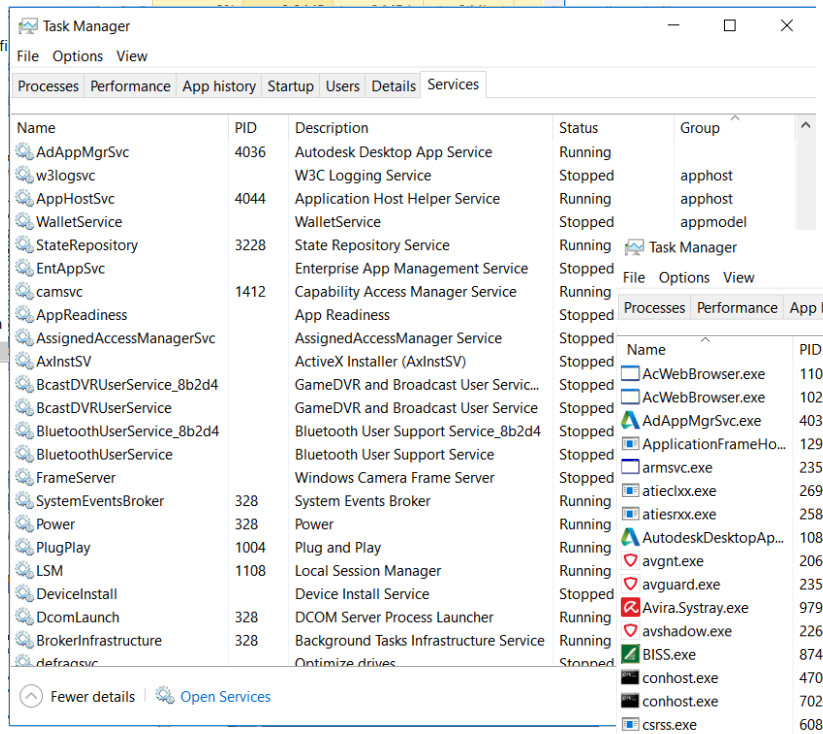
Приложен слой

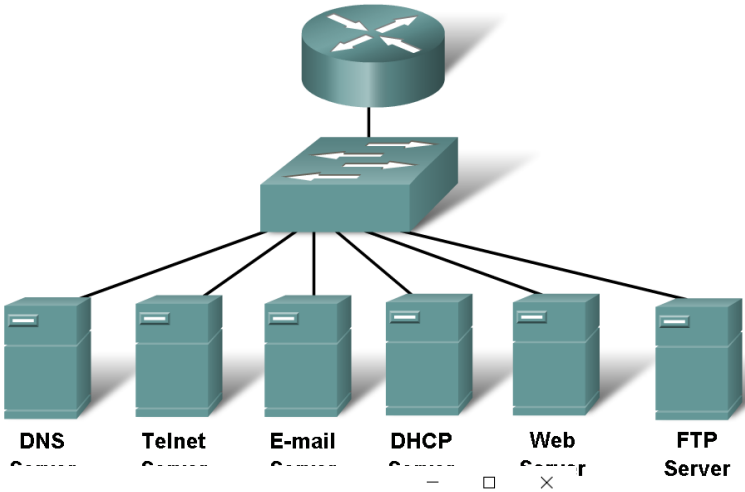
- Поддържа приложенията и крайните потребителски процеси. Приложното ниво на TCP/IP модела включва горните три нива на OSI модела.

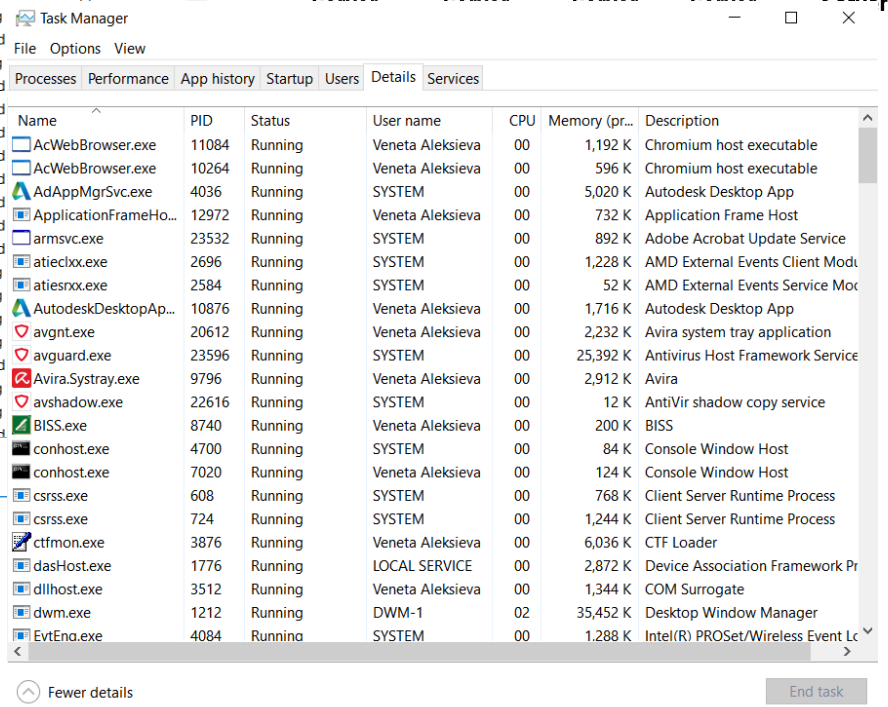


Услуги и протоколи на приложния слой



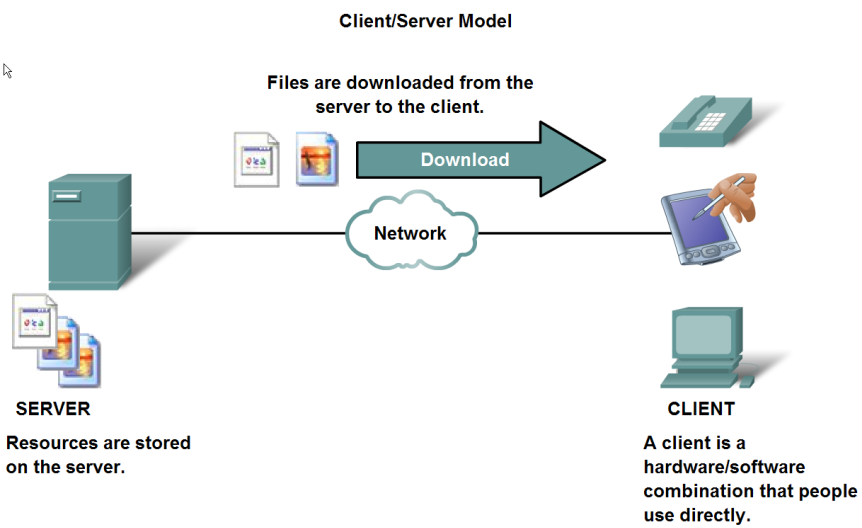


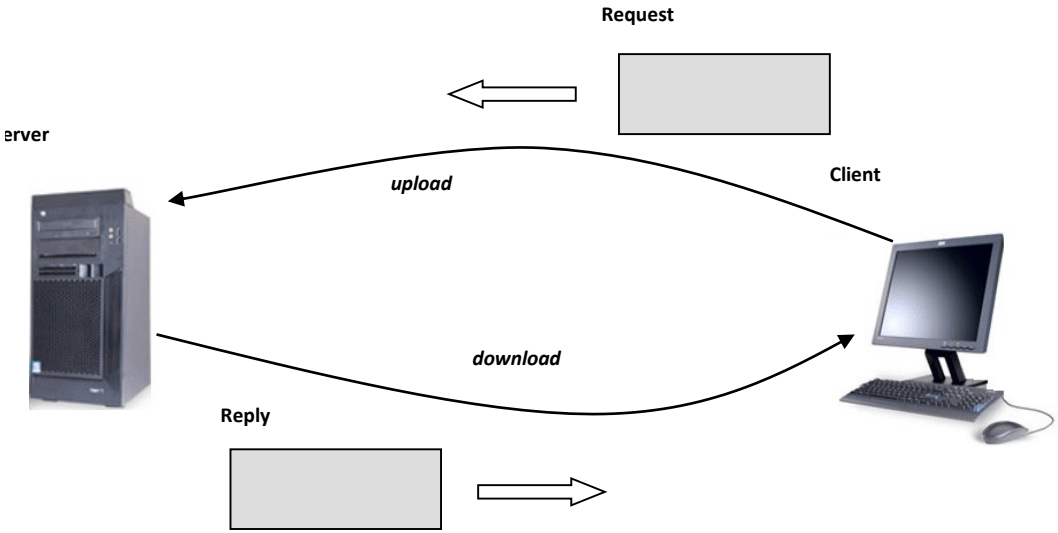




Модел клиент-сървър

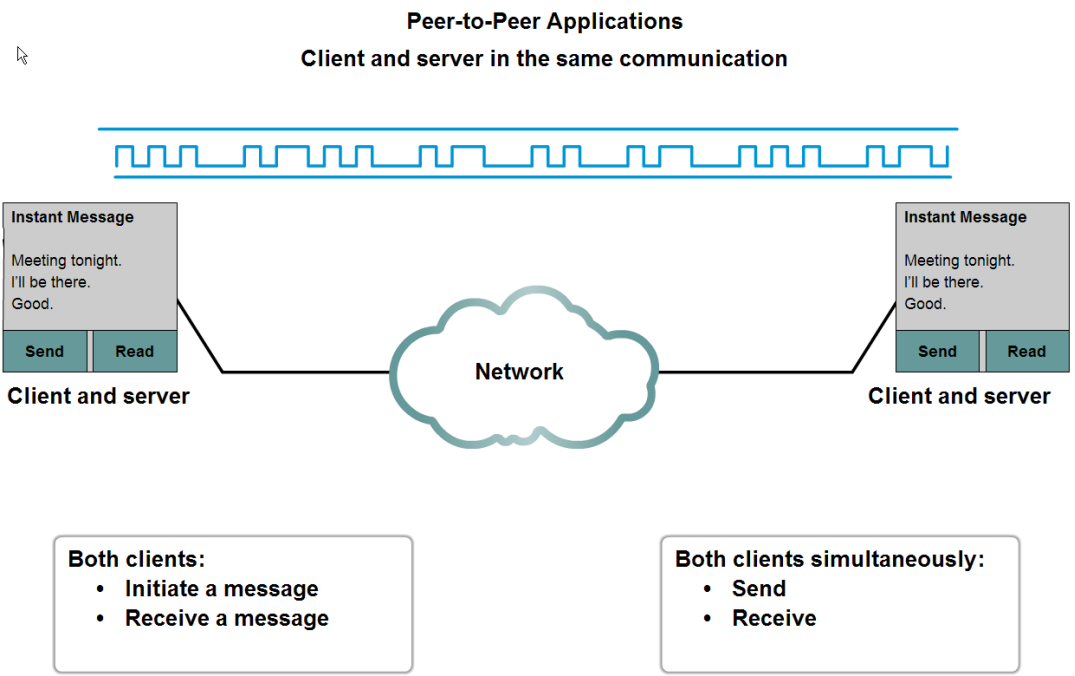
- Софтуерът, изискващ информация се нарича „клиент”
- Софтуерът, отговарящ на заявка за информация се нарича „сървър”
- Сървърът съхранява и предоставя ресурси на клиентите
- Предаването на данни от клиента към сървъра се нарича „upload”
- Предаването на данни от сървъра към клиента се нарича „download”

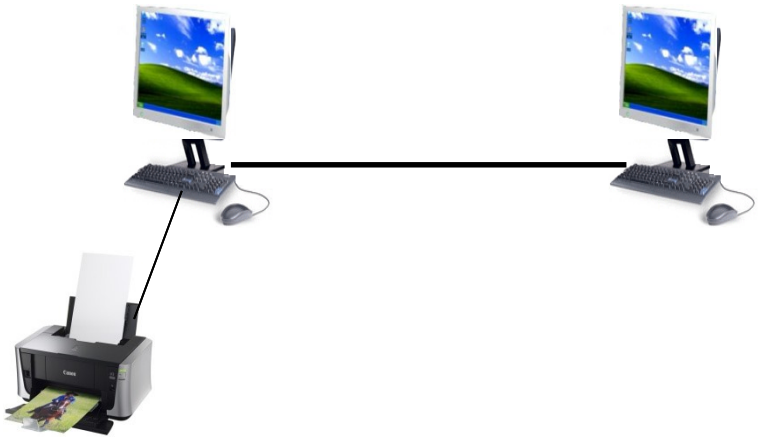




Модел Peer-to-Peer

- Два или повече свързани в мрежа компютри, които споделят ресурси (файлове или принтери)
- Няма необходимост от отделен сървър
- Типичен пример – Windows мрежа





DNS- Динамично резолване на име на хост

Статично резолване на име на хост - Файлът HOSTS преди

- ARPANet - текстов файл **HOSTS.TXT**
  - имената на машините и техните IP адреси
  - поддържал се е от информационния център (NIC) в Станфордския Научен Институт (SRI)
  - Мрежовите администратори е трябвало да получават ежеседмично обновени копия на този файл.
  - Мрежовите администратори е трябвало да изпращат информация за всяка добавена и премахната машина от техните мрежи.
- Неефективност:
  - Нарастването размера на файла водело до проблеми с трафика при неговото разпространение;
  - Налице е била възможността от дублиране на имената от различните администратори;
  - Невъзможност за бързо отразяване на промените в компютърните мрежи.

Файлът HOSTS сега

- Остава като алтернатива на резолване на имена към IP адреси вместо DNS
- Под прекия контрол на администратора в локалния компютър, за разлика от DNS резолверите.
- В Windows 10 е в System32/drivers/etc/hosts
- В Linux е в /etc/hosts

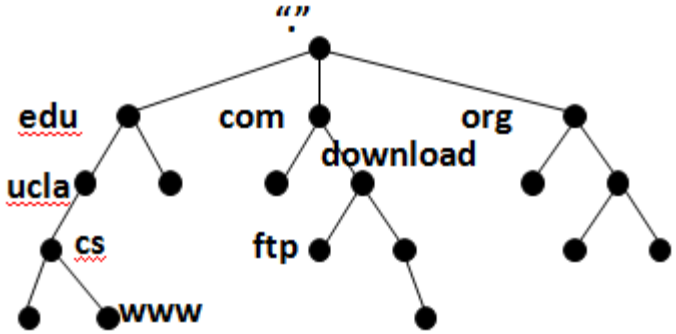
```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
#       127.0.0.1       localhost
#       ::1             localhost
```

Domain Name System (DNS)

- Домейн - област, територия, сфера, обсег.
- Система, описана за първи път през 1983 г. и реализирана през 1984 г. (RFC1034, RFC1035)
- Динамично резолване на име на хост.
- Базира се на разпределена база от данни, реализираща йерархическа система за именуване.
- Най-разпространената съвременна реализация е на Университета в Бъркли – *Berkeley Internet Name Domain* (BIND), която се включва в почти всички дистрибуции на Unix (*www.isc.org/sw/bind*).

Пространство на имената

- Логическа йерархична дървовидна структура
- Всеки възел е DNS име (до 63 символа)
- Коренът на дървото (*root)* е “**.**”
- До 127 поднива
- Имената на дъщерните възли на един и същ родител в дървото трябва да имат различни имена.
- Крайните възли (*листа*) в дървовидната структура представят винаги имена на хостове, докато междинните възли могат да указват както хост, така и домейн.

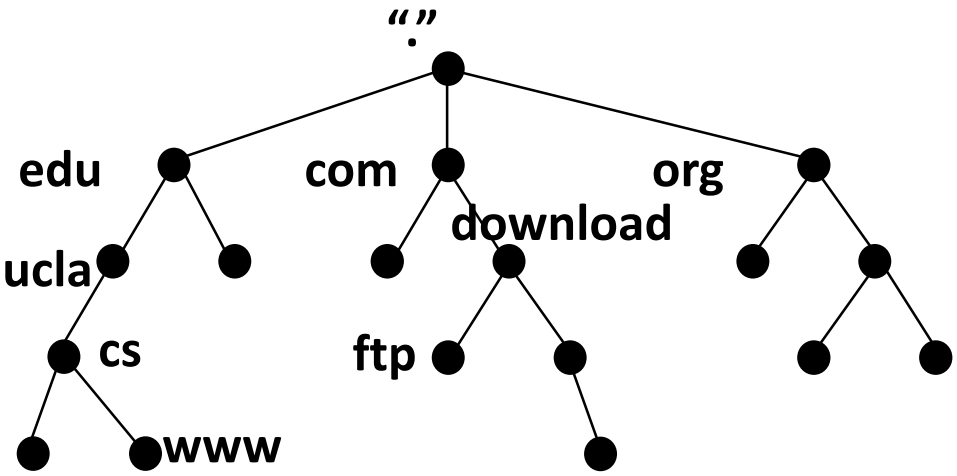


Fully Qualified Domain Name – FQDN

- Пълното домейн име (FQDN) на всеки възел се формира като последователност от имена по пътя от възела до корена, разделени с точка.
- Имената се записват от по-значимото (името на машината) към по-малко значимото (корена).
- Например:

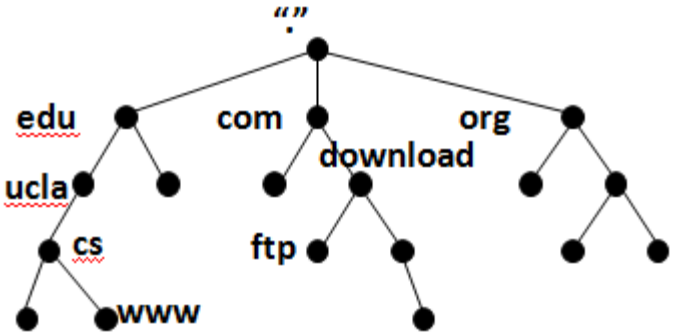
www.cs.ucla.edu

ftp.download.com



Top Level Домейни

- Домейните от най-високо ниво са TLDs (Top Level Domains)
- Те са дефинирани от ICANN.
- В Интернет не могат да съществуват два напълно еднакви домейна.
- Всеки домейн може да се раздели на поддомейни.



Видове top level домейни

- **Организационни домейни** (оригинални TLD) - 3-символни обозначения

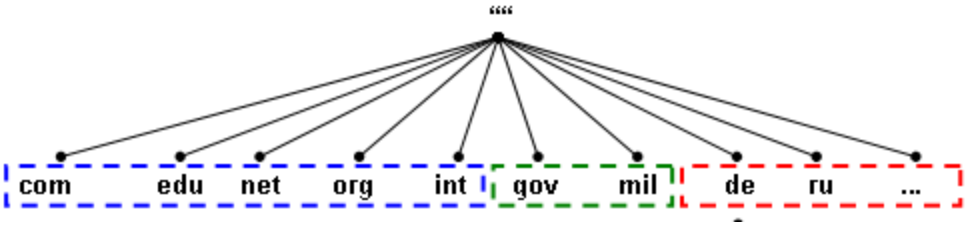
.com, .org, .net и др.;

- **Географски домейни** (национални TLD)- 2-символни обозначения, съобразно ISO 3166

.bg, .ru, .es и др.;

- **Обратен домейн (**инфраструктурни TLD**)**.

.arpa и др.;



- **Нови TLD**

.archi, .info, .engineer и др.

Българските домейни: .bg

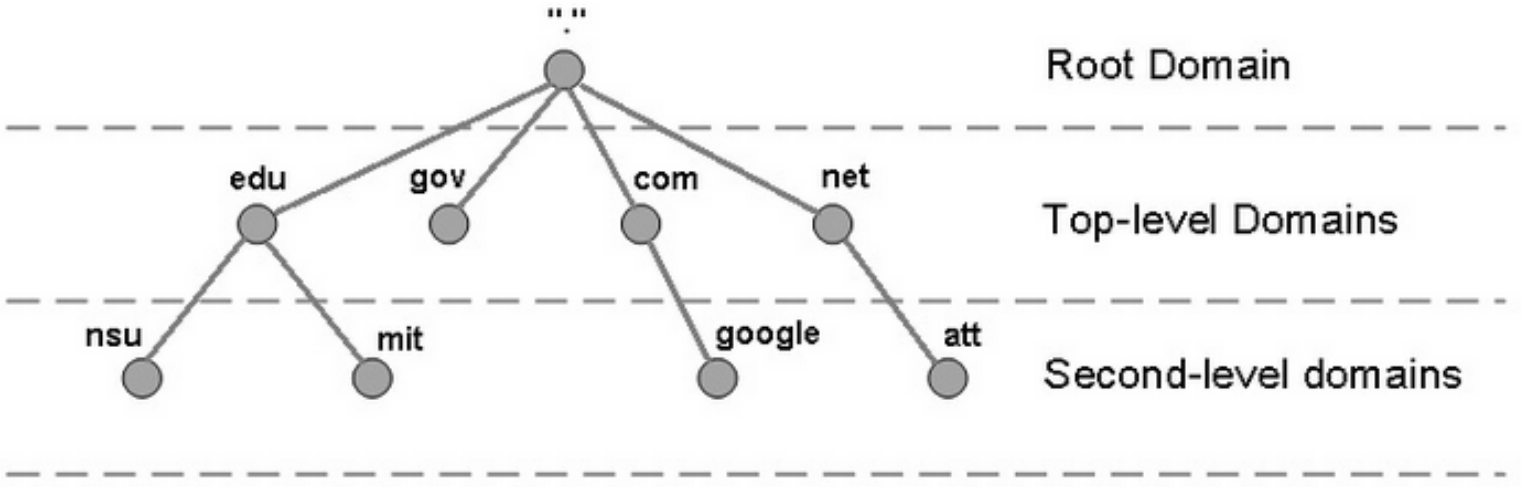
- 1991 година се регистрира от “Цифрови Системи”
- 2001 година се поема от “Регистър.БГ”
- 2006г. “Регистър.БГ” започва да предлага домейни в области от второ ниво.
  - Новите области са: .a.bg, ..., .z.bg, .0.bg, ..., .9.bg.
- От 05.09.2009 година към областта .bg могат да се регистрират домейни с имена на кирилица (Internationalized Domain Names – IDNs).
  - Изискване - името да се изписва само на кирилица и да съдържа поне една буква, отличаваща се визуално от латинската азбука: б, г, д, ж, и, й, л, п, ф, ц, ч, ш, щ, ь, ъ, ю, я.
- Примери:
  - http://хамали.bg/;
  - http://www.лещи.bg/.

Българските домейни: .бг

- Към текущия момент “Регистър.БГ” е част от “ИМЕНА.БГ” АД , които поддържат регистрацията на Интернет имена в областта **.бг** (на кирилица).
- Областта **.бг** не е първата област на кирилица.
- Областта **.рус** вече съществува на кирилица.
- Причината за забавянето на разрешението за използване на домейна **.бг** е визуалната сходност между символите **.бг** и **.br** (националният TLD на Бразилия).

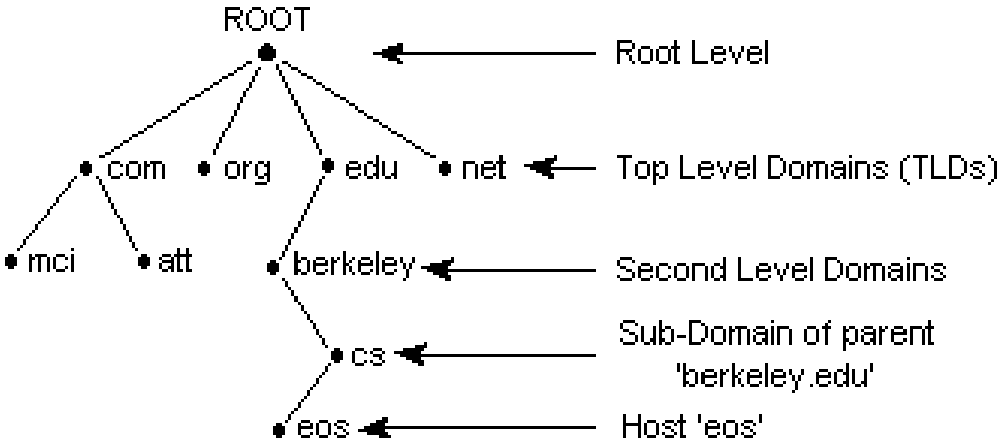
Домейни от следващи нива

- Домейните след тези от най-високо ниво се наричат *домейни от първо ниво* (*first-level domains*).
- След тях следват домейните от второ ниво и т.н.
- Дефинирането на поддомейните се подчинява на различни логически или организационни принципи на разделяне на домейните от по-горните нива.
- Пример: cs.tu-varna.bg



Делегиране

- Процесът на предоставяне на права за управление на разпределените бази данни е известен под наименуването ***делегиране* (*delegation*).**
- Всеки поддомейн може да бъде делегиран на отделна организация:
  - Тя отговаря за коректната информация в нейния домейн
  - Тя може да разделя домейна си на нови поддомейни и да делегира права за тяхното управление на други организации.

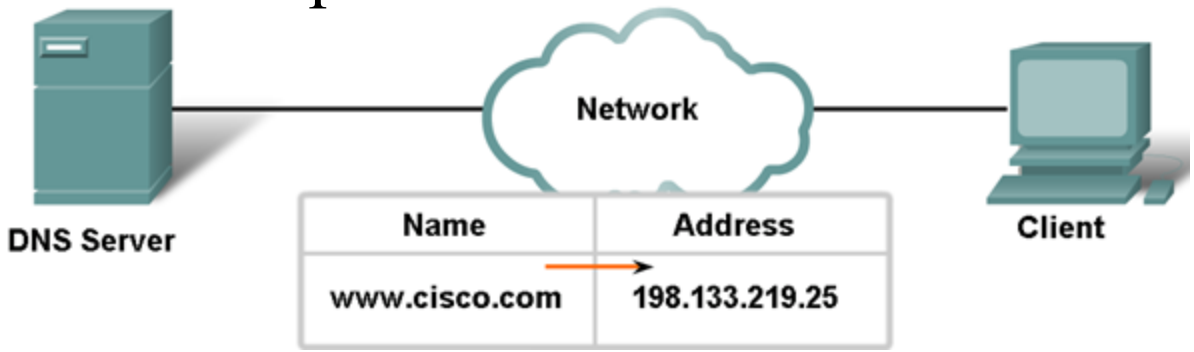


Принцип на действие на DNS сървърите

- DNS сървърът получава заявка за разрешаване.
- DNS сървърът търси изискваната информация от базата данни за неговата зона.
- Ако информация в БД липсва, той търси в:
  - собствената си кеш памет,
  - от други DNS сървъри,
  - насочва клиента към друг DNS сървър, който може да знае отговора.

Разрешаване на имената

- Получаването на съответствията имена – IP адреси се реализира посредством процес, известен под наименуването *разрешаване* (*resolving*).
- Сървърите на имената имат задачата да открият това съответствие не само за зоните, за които са отговорни, но и търсейки в цялото пространство на имената на данни, за които не се явяват пряко отговорни.
- В DNS се използват два основни типа запитвания: рекурсивни и итеративни.

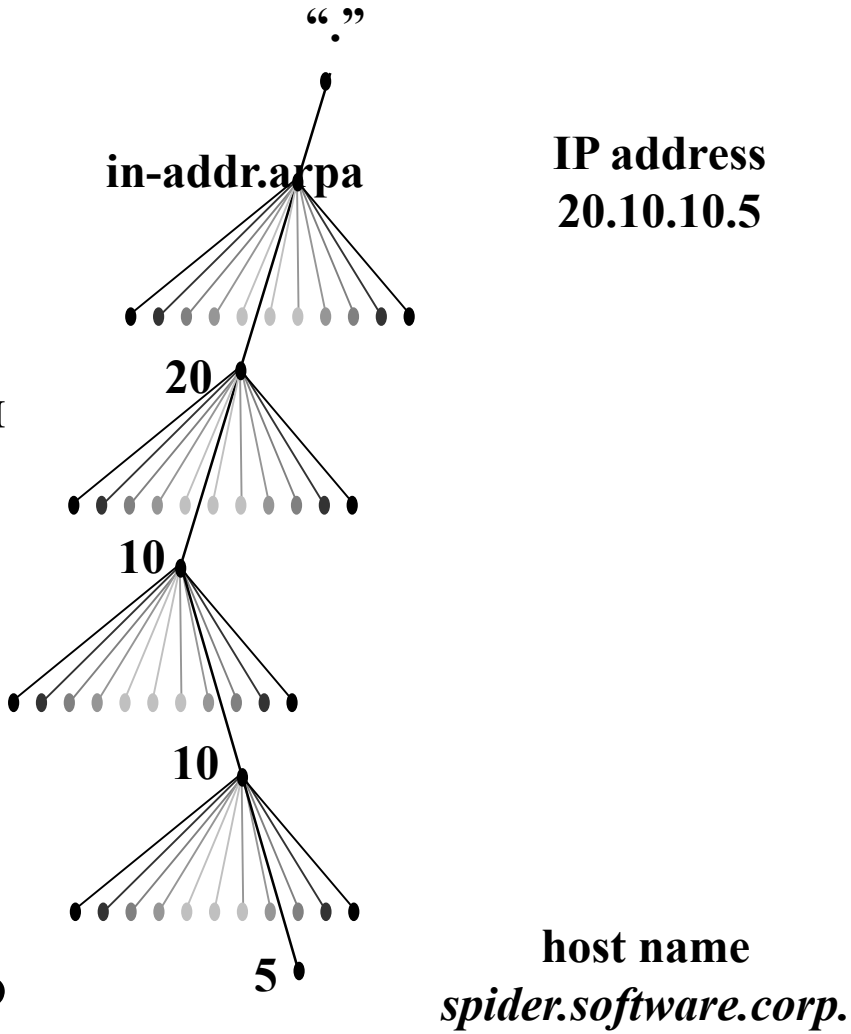


Обратно резолване

- Обратно разрешаване – по даден IP адрес да се получи Интернет името на хоста.
- Това се използва от програми, реализиращи контрол на достъпа, базиран на FQDN или за диагностика и откриване на проблеми в мрежовите комуникации.
- Тъй като данните, съдържащи IP адресите са индексирани в базите на DNS сървърите, търсенето на име е значително ускорено.
- Обратното търсене, обаче е неефективно.

Домейнът in-addr.arpa

- Специален фиктивен домейн с името **in-addr.arpa**.
- Възлите в този домейн се именуват чрез обратно подреждане на числата, формиращи IP адреса, представен в десетичен вид с точки.
- **in-addr.arpa.** може да има до 256 поддомейни, по един за всяка възможна стойност от първия октет на IP адреса, всеки от тях - до 256 поддомейни, съответстващи на стойностите от втория октет на IP адреса и т.н. до последния октет.



DNS и транспортен слой

- DNS ползва номер на порт 53
- Работи и с UDP, и с TCP

DHCP- динамично задаване на адреси на хостове

Адресиране в мрежата

Всяко устройство в мрежата се нуждае от адрес, който е получен:

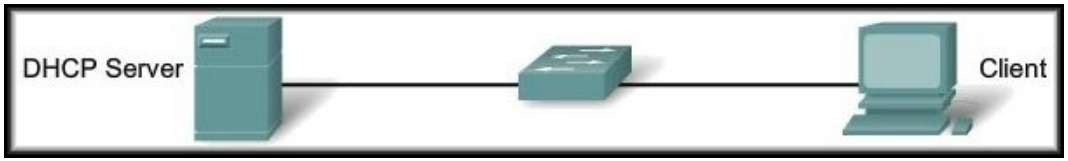
- **Статично** - за маршрутизатори, сървъри, които няма да сменят своето място в мрежата.
- **Динамично** – за Desktop клиенти и работни станции с адреси от една IP subnet мрежа.

Предимства на динамичното адресиране

- Администраторите предпочитат DHCP услугите да се предлагат от мрежов сървър.
  - Улеснява разрастването на мрежата.
  - Улеснява управлението и администрирането й.
- За SOHO решението е маршрутизатор, който е конфигуриран да осигурява DHCP услуги, за да не се закупува скъп сървър специално за това.

DHCP

- Dynamic Host Configuration Protocol (DHCP)
- Заменя Bootstrap Protocol (BOOTP)
- За IPv4 - RFC 1531 от 1993г., обновен от RFC 2131 през 1997г.
- За IPv6 - (DHCPv6) -RFC 3315 от 2003г.



Формат на DHCP съобщението

- Има съвместимост между DHCP и BOOTP.

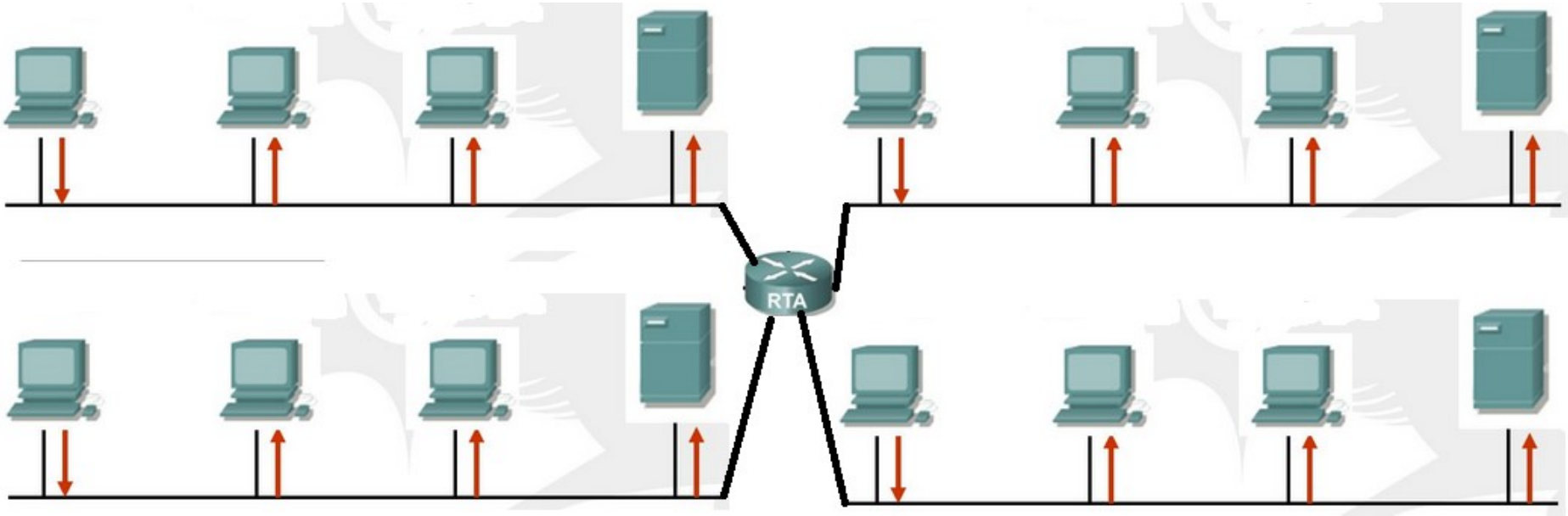
Същите като BOOTP

8	16	24	32
OP Code (1)	Hardware type (1)	Hardware address length (1)	Hops (1)
Transaction Identifier			
Seconds – 2 bytes		Flags – 2 bytes	
Client IP Address (CIADDR) – 4 bytes			
Your IP Address (YIADDR) – 4 bytes			
Server IP Address (SIADDR) – 4 bytes			
Gateway IP Address (GIADDR) – 4 bytes			
Client Hardware Address (CHADDR) – 16 bytes			
Server name (SNAME) – 64 bytes			
Filename – 128 bytes			
DHCP Options – variable			

Добавени, за да поддържат DHCP функциите.

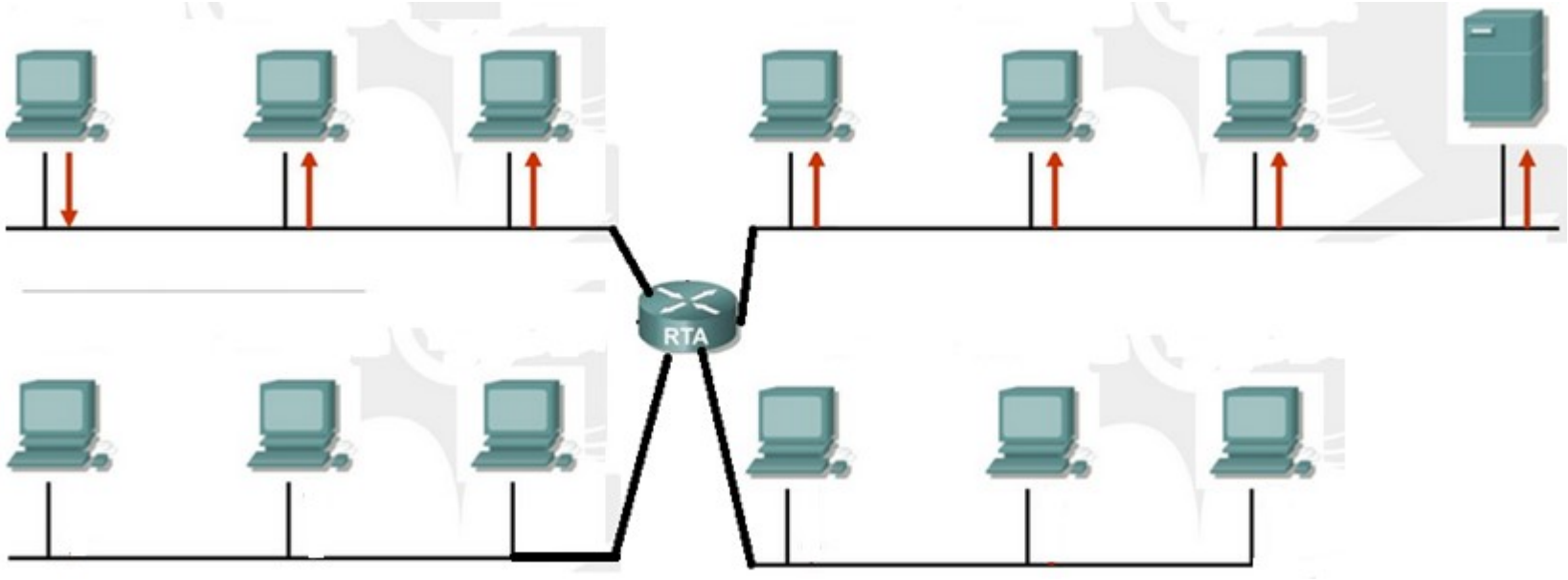
BOOTP

- Трябва да има BOOTP сървър във всеки сегмент.



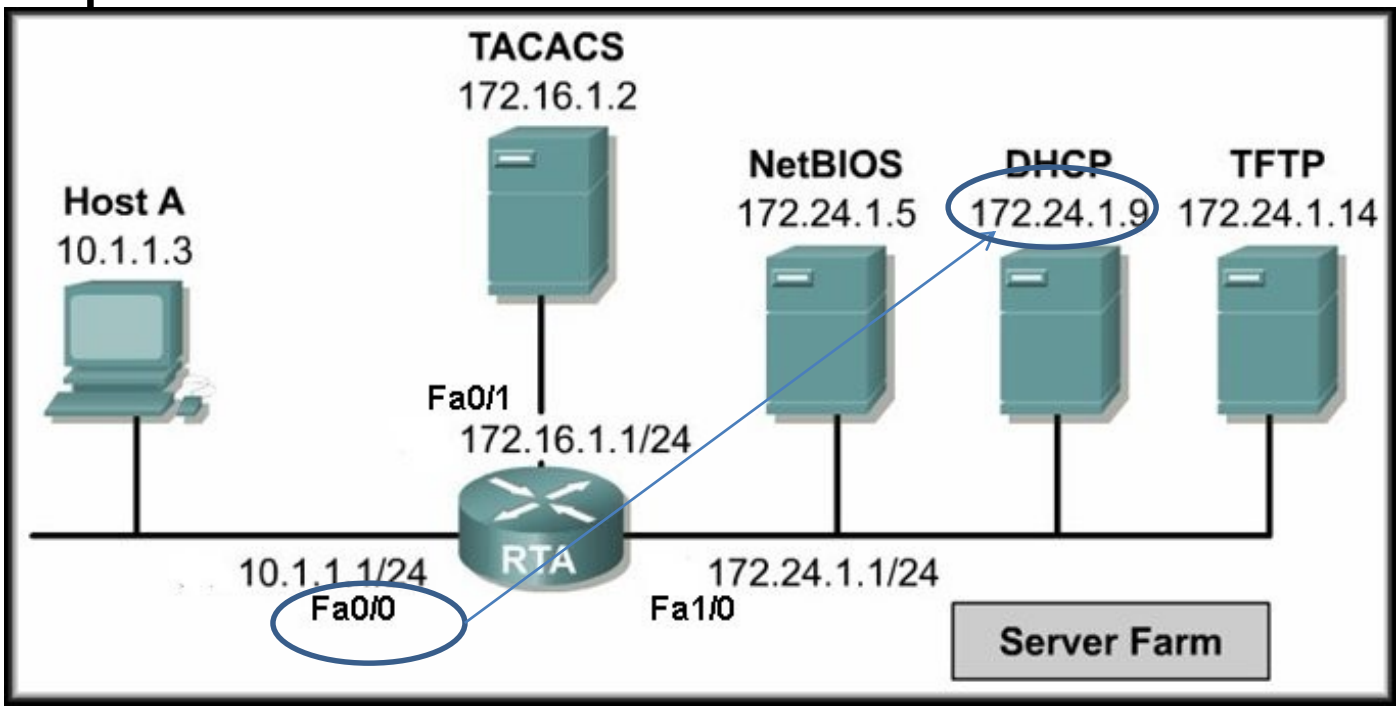
DHCP

- Един DHCP сървър е достатъчен за всички устройства във всички сегменти.
- Но как минава бродкаста през маршрутизатора?

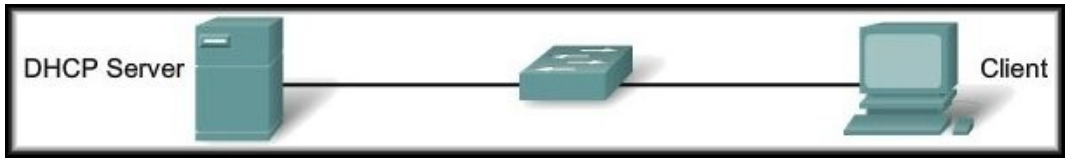


DHCP Relay

DHCP Relay - чрез конфигуриране на интерфейса (fa0/0) на помощен адрес (172.24.1.9) на маршрутизатора ще се препращат DHCP broadcasts до правилния сървър.

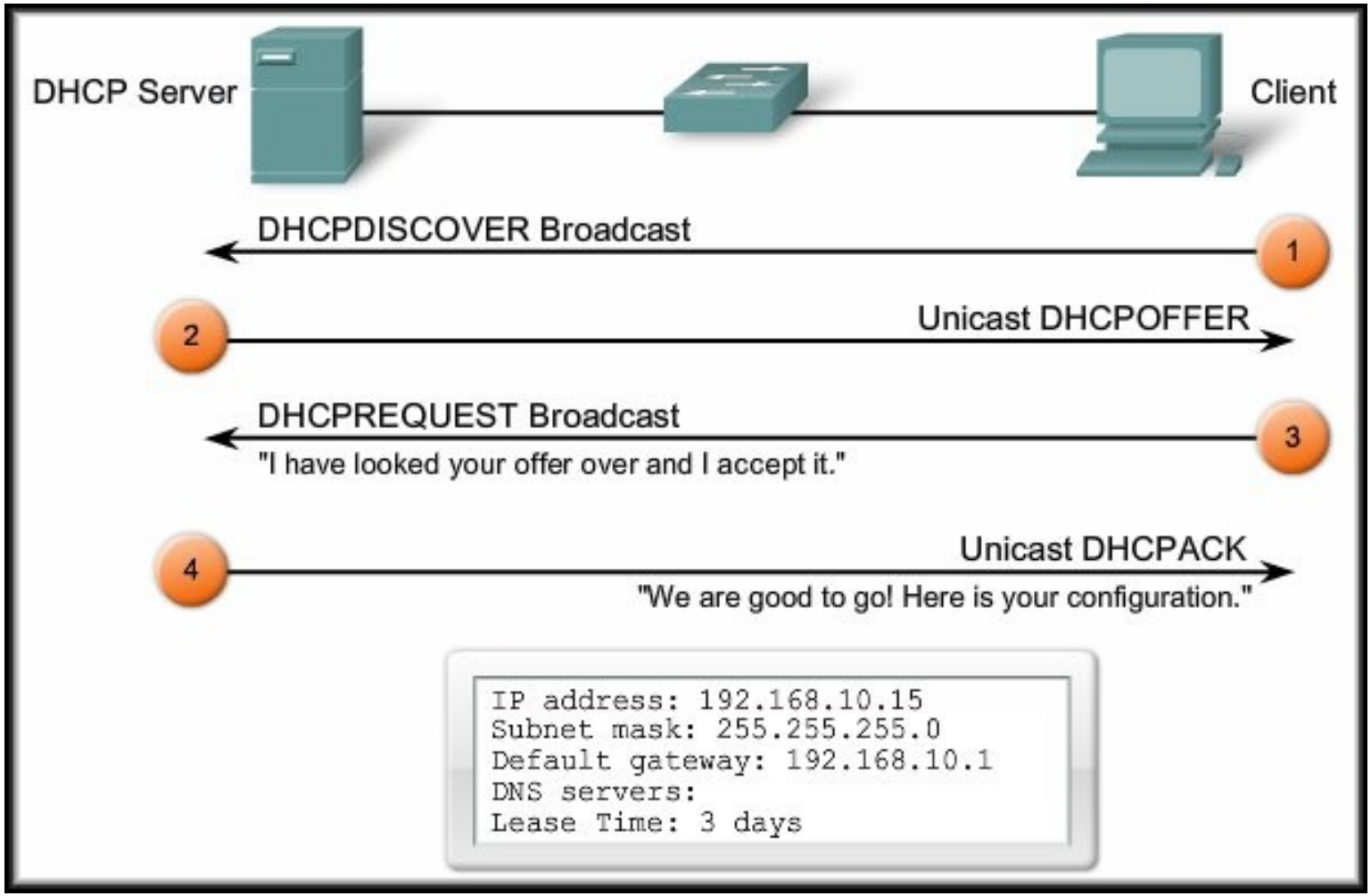


Разпределяне на адреси с DHCP



- **Ръчно**
  - IP адресите за клиентите предварително са разпределени от администратора (по MAC адреси) и DHCP само предава адреса на клиента.
- **Автоматично** (при DHCPv6)
- **Динамично**
  - Клиентът се свързва с DHCP сървъра
  - Сървърът му присвоява или отдава IP адрес за ограничен период от време
  - Клиентът регистрира информацията и изпраща ARP заявка, за да провери уникалността на адреса
  - Клиентът се свързва в мрежата с така получения IP адрес до изтичане на периода за отдаване.
  - Клиентът трябва периодично да се обръща към DHCP сървъра, за да удължава периода на ползване на адреса.
  - Адрес, който е отдаден, но не е използван след периода на отдаване се връща в pool-а, за да бъде отдаден на друго устройство.

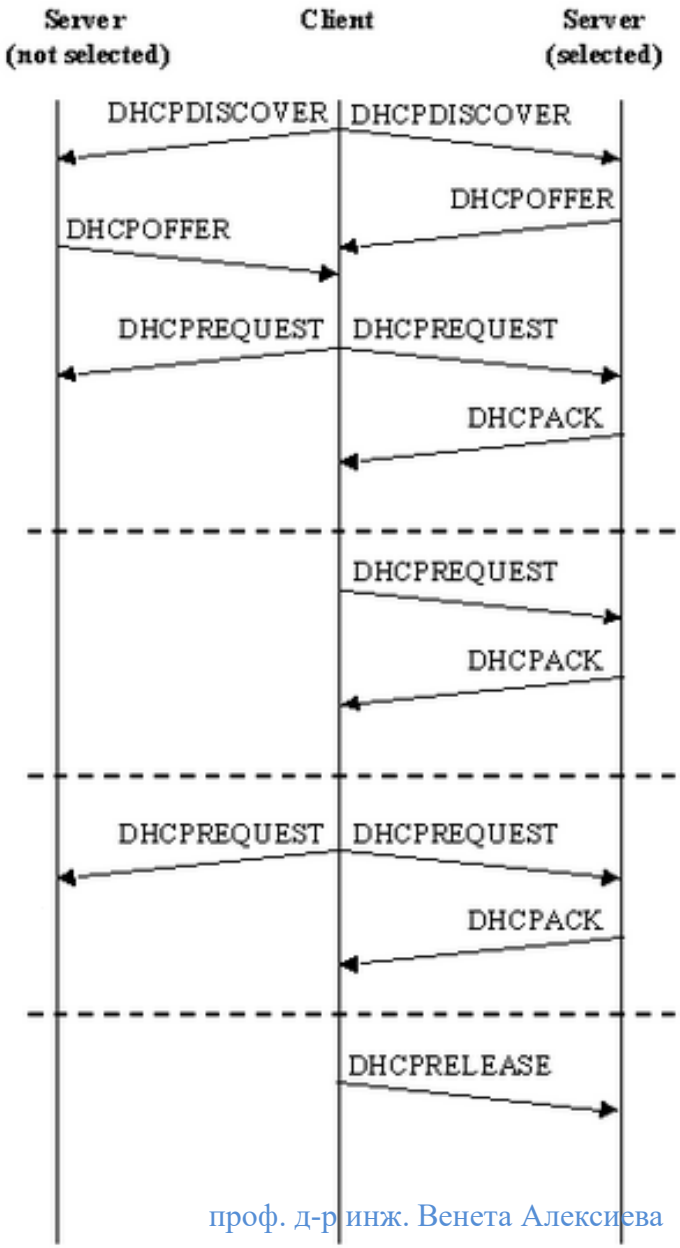
DHCP съобщения



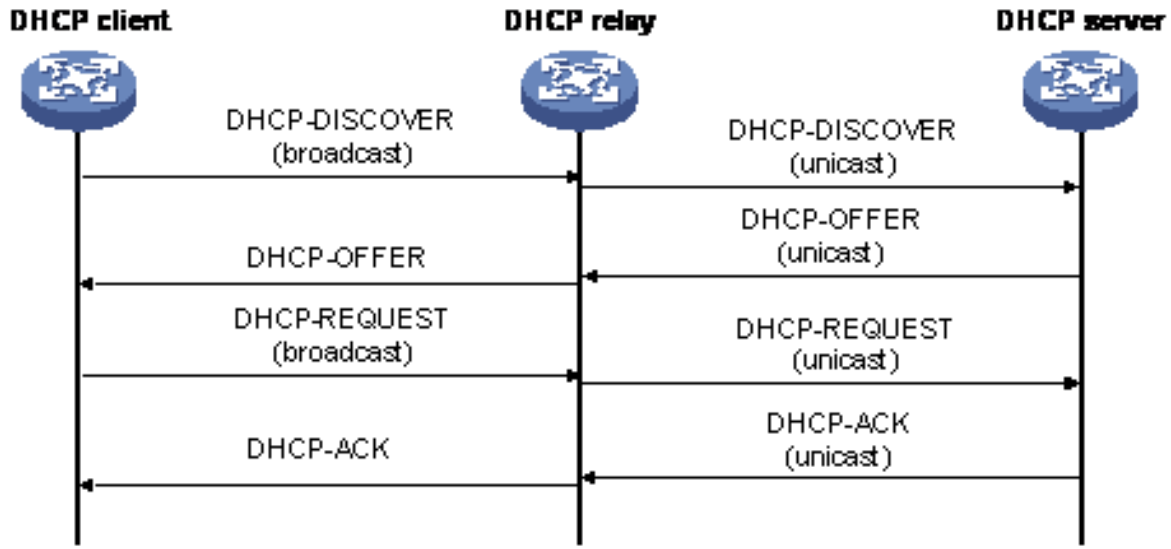
А конфликтите?

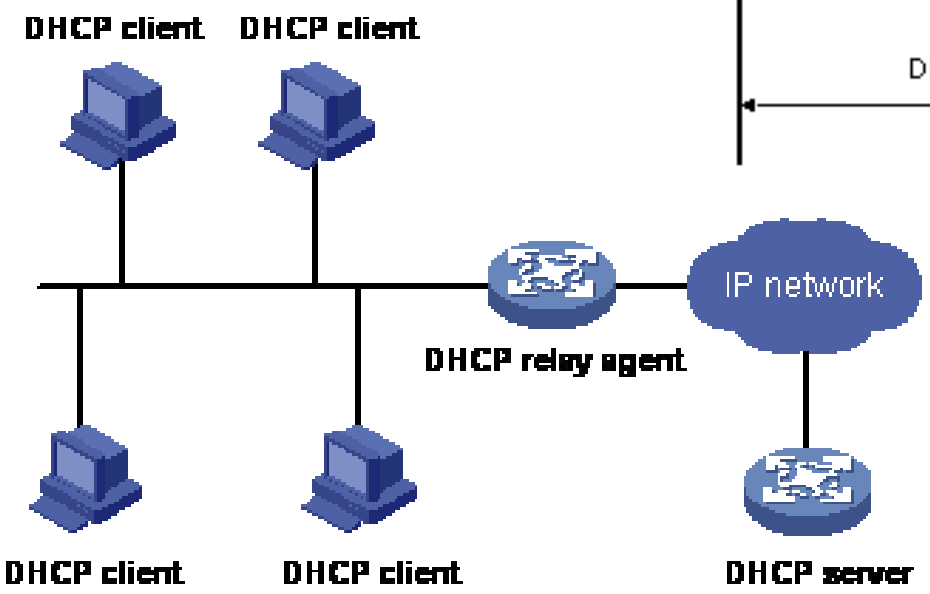
- По подразбиране DHCP сървърът ping-ва адресите от pool-а преди да отдаде адрес на клиента.
- Ако ping-а не получи отговор в 500 ms (times out), DHCP сървърът предполага, че адресът не е в употреба и го отдава.

А при 2 DHCP сървъра?



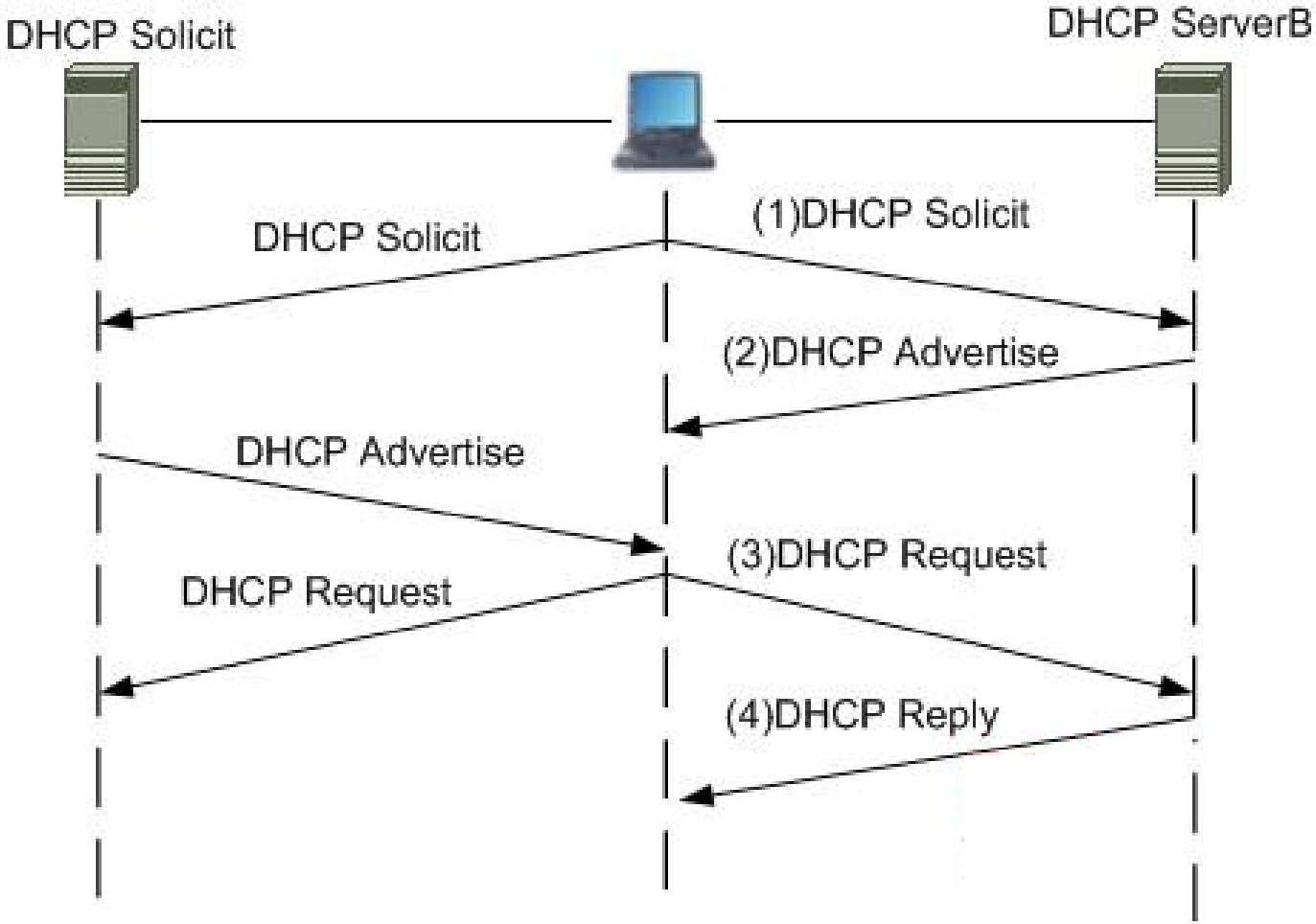
DHCP Relay агент





- DHCP Relay агентът не се инсталира на машината, на която е DHCP сървърът.
- FF02::C е IPv6 адресът за DHCP Servers / Relay Agents.

DHCPv6


DHCP Solicit
DHCP ServerB
DHCP Solicit
(1)DHCP Solicit
(2)DHCP Advertise
DHCP Advertise
(3)DHCP Request
DHCP Request
(4)DHCP Reply

DHCPv6 / DHCPv4

DHCPv6 Message Types	DHCPv4 Message Types
SOLICIT(1)	DHCPDISCOVER
ADVERTISE(2)	DHCPOFFER
REQUEST (3), RENEW (5), REBIND (6)	DHCPREQUEST
REPLY (7)	DHCPACK/DHCPNAK
RELEASE (8)	DHCPRELEASE
INFORMATION-REQUEST (11)	DHCPINFORM
DECLINE (9	DHCPDECLINE
CONFIRM (4)	None
RECONFIGURE (10)	DHCPFORCERENEW
RELAY-FORW (12), RELAY-REPLY (13)	None

Stateful DHCPv6

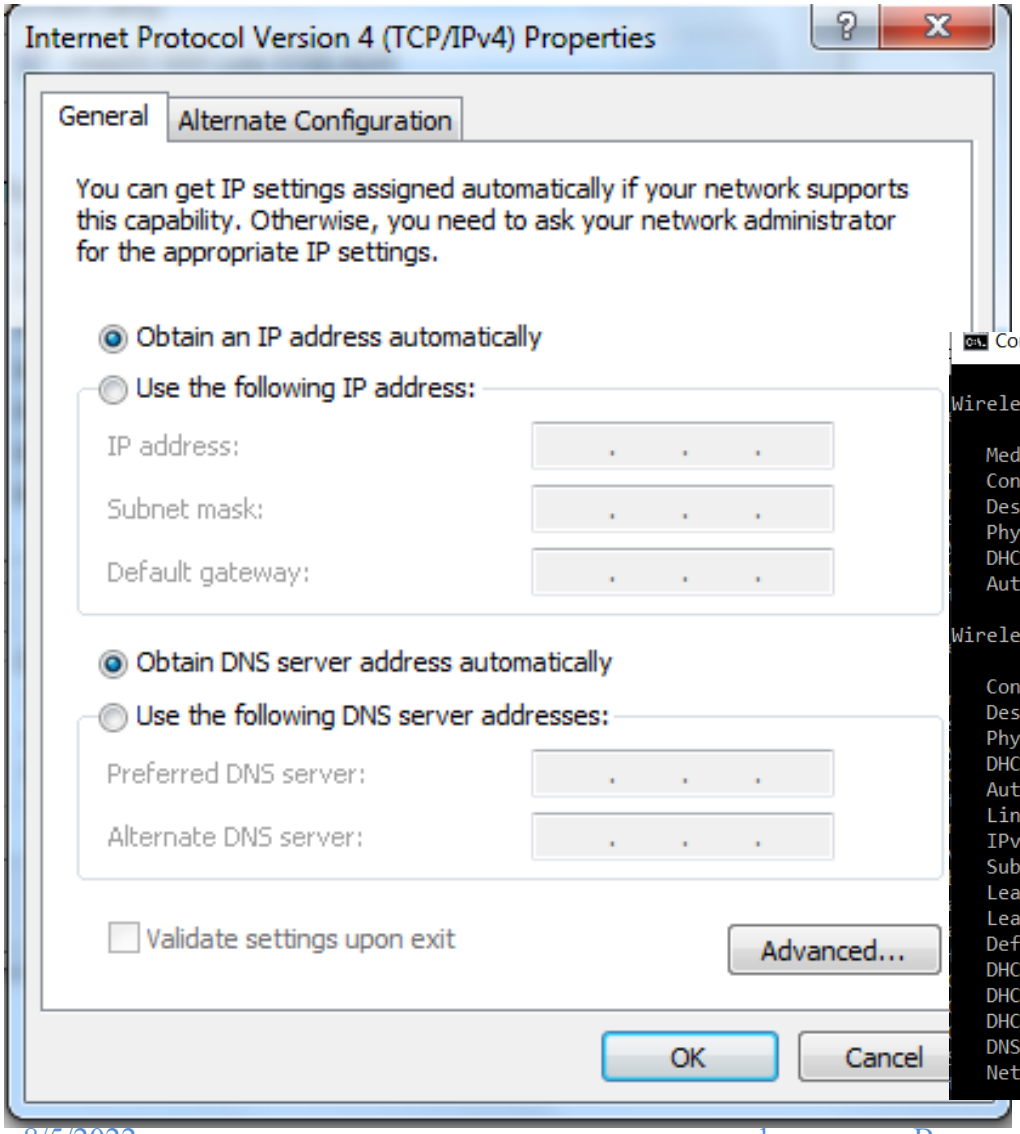
- При Stateful DHCP присвояването на адреси се управлява централно.
- DHCPv6 се прилага по два начина
  - **Rapid Commit (трябва да е разрешено и на сървъра, и клиента)**- DHCP клиентът взема конфигурацинните параметри от сървър чрез 2 съобщения – solicit и reply.
  - **Normal Commit (по подразбиране)** - DHCP взема конфигурационните параметри от сървър чрез 4 съобщения -solicit, advertise, request and reply.

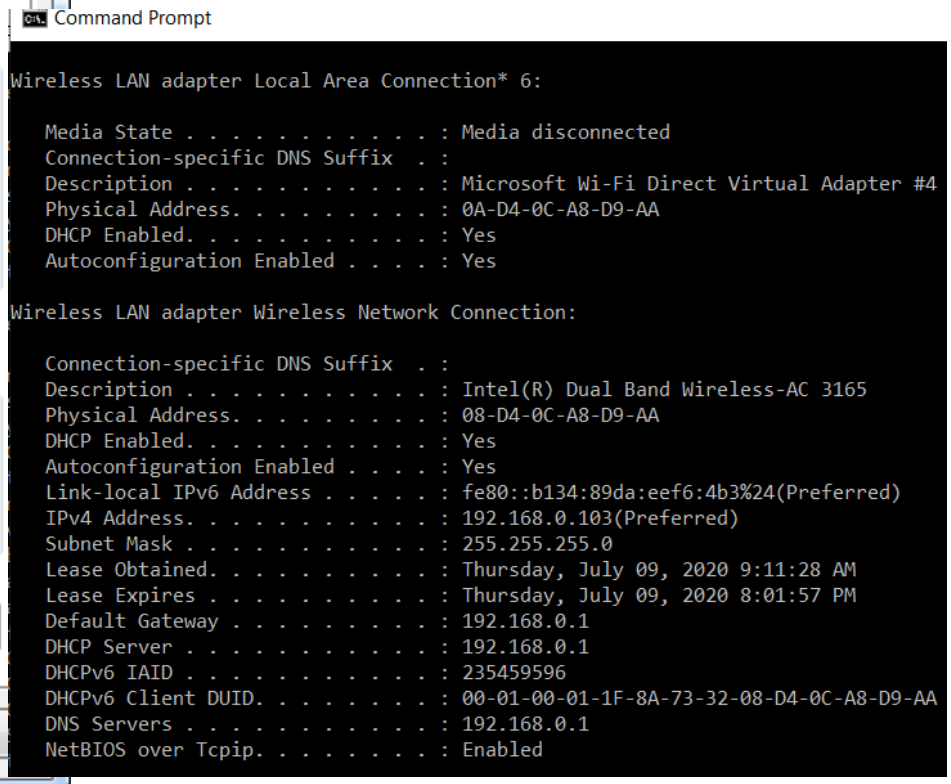
Конфигуриране на DHCP клиент под Windows

ipconfig/all

ipconfig/release

ipconfig/renew





APIPA адреси

- *Automatic Private IP Addressing* (APIPA) - ако клиентът не може да получи конфигурация, се конфигурира самостоятелно с IP адрес от диапазон, резервиран от Microsoft.
- От 169.254.0.1 до 169.254.255.254, с маска 255.255.0.0.
- Периодично (на 5 минути) клиентът проверява за наличие на DHCP сървър в локалния сегмент.
- При откриване на DHCP сървър, APIPA адресът се освобождава и се приема DHCP конфигурацията.

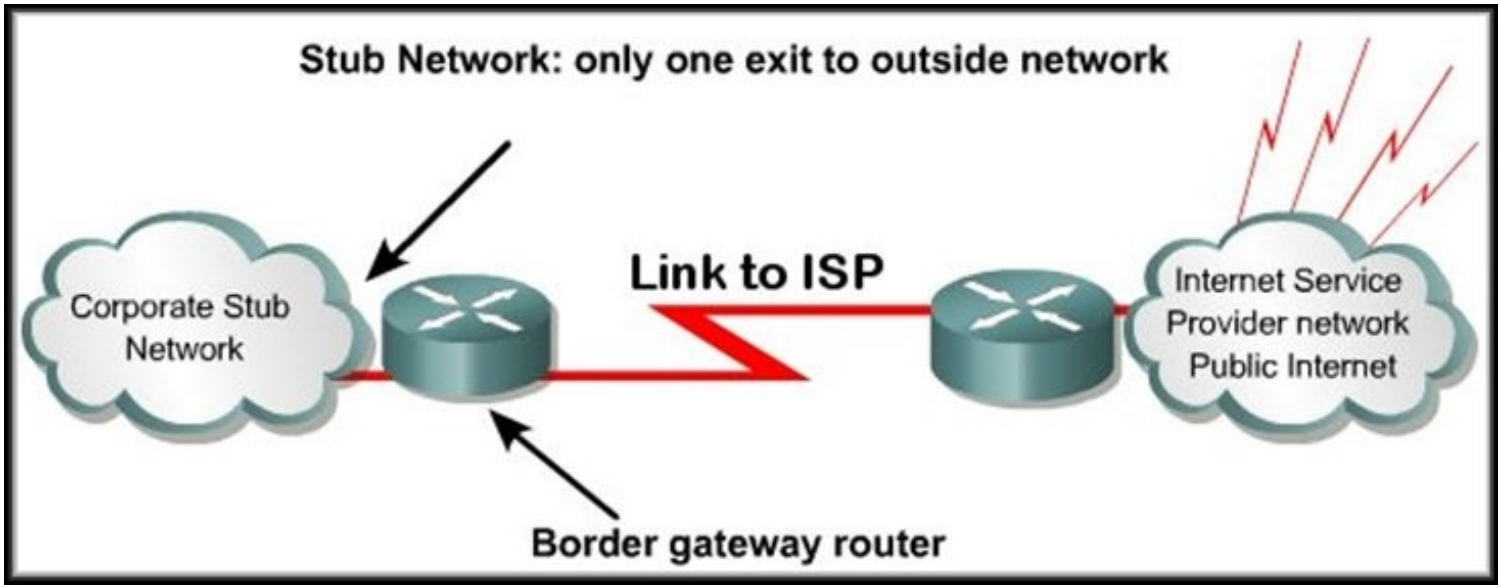
NAT- преобразуване на адреси

Същност на NAT

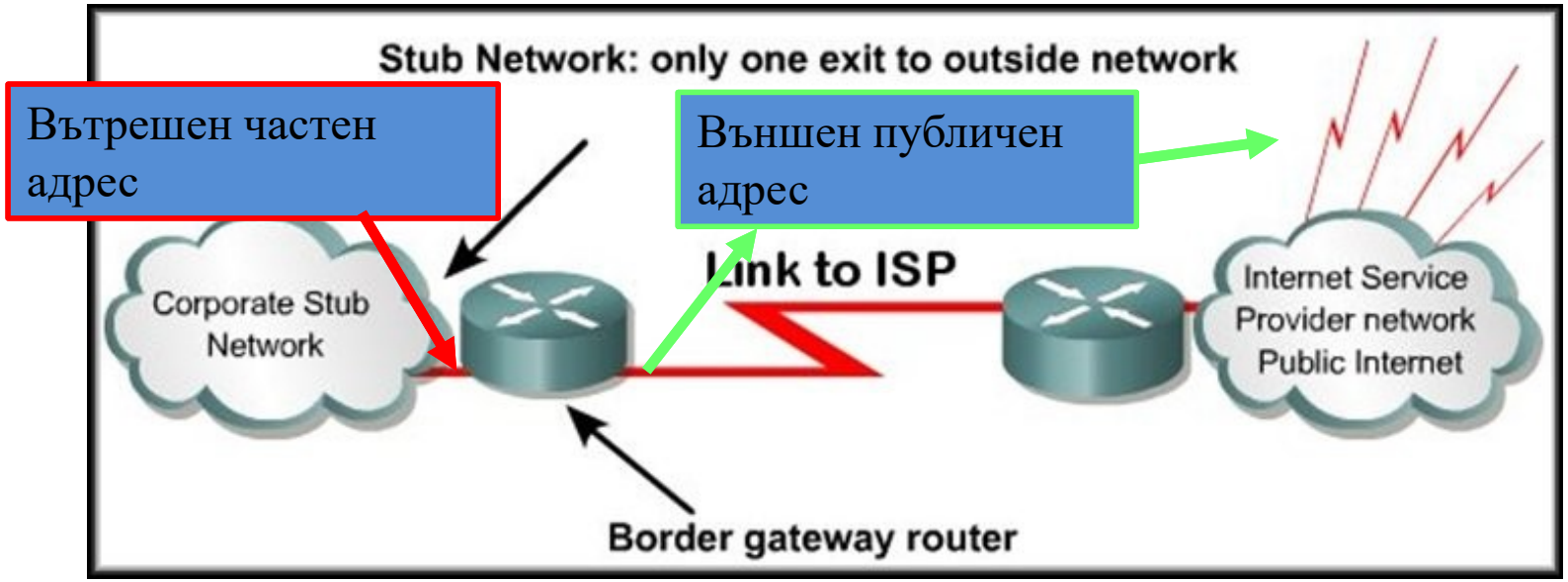
- NAT преобразува вътрешния IP адрес на клиента във външен адрес.
- Когато е конфигуриран NAT на маршрутизатора, той има >=1 публични адреса за външната мрежа.
- За външните потребители, целият трафик от тази мрежа изглежда сякаш идва от 1 адрес (или pool).

Stub мрежа

- Stub мрежа е мрежа, която има само 1 връзка към нейната съседна мрежа.
- NAT работи на гранично устройство между тези мрежи.

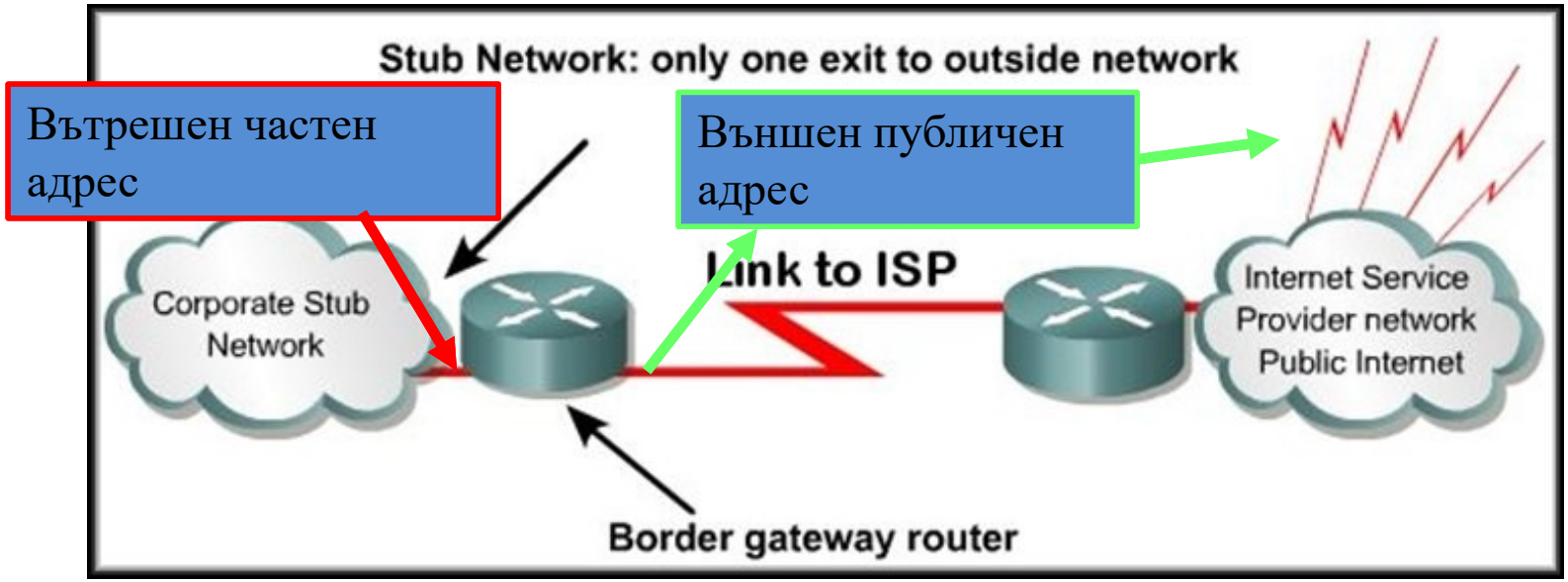


Действие на NAT



- Щом хост от вътрешната мрежа иска достъп до хост от външната мрежа, пакета се изпраща на граничния маршрутизатор.
- Граничният маршрутизатор (gateway) изпълнява NAT и преобразува вътрешния частен адрес във външен публичен адрес.

Действие на NAT



- Преобразуването става с помощта на вътрешна таблица за транслиране.
- Съдържанието й зависи от типа на преобразуването, което е избрано: static NAT, dynamic NAT или Port Address Translation (PAT).

Предимства на NAT

- Пести брой на нужните публични адреси.
- Увеличава гъвкавостта на връзките в публичната мрежа.
- Осигурява съгласуваност между адресите от вътрешната и външната мрежа.
- Осигурява мрежова сигурност.

Недостатъци на NAT

- Производителността на мрежата намалява.
- Функционалността End-to-end намалява.
- Губи се проследяването на End-to-end.
- Тунелирането е по-сложно за създаване.
- Създаване на TCP конекции може да се прекъсне.
- Може да се наложи мрежовата архитектура да се изгради наново.

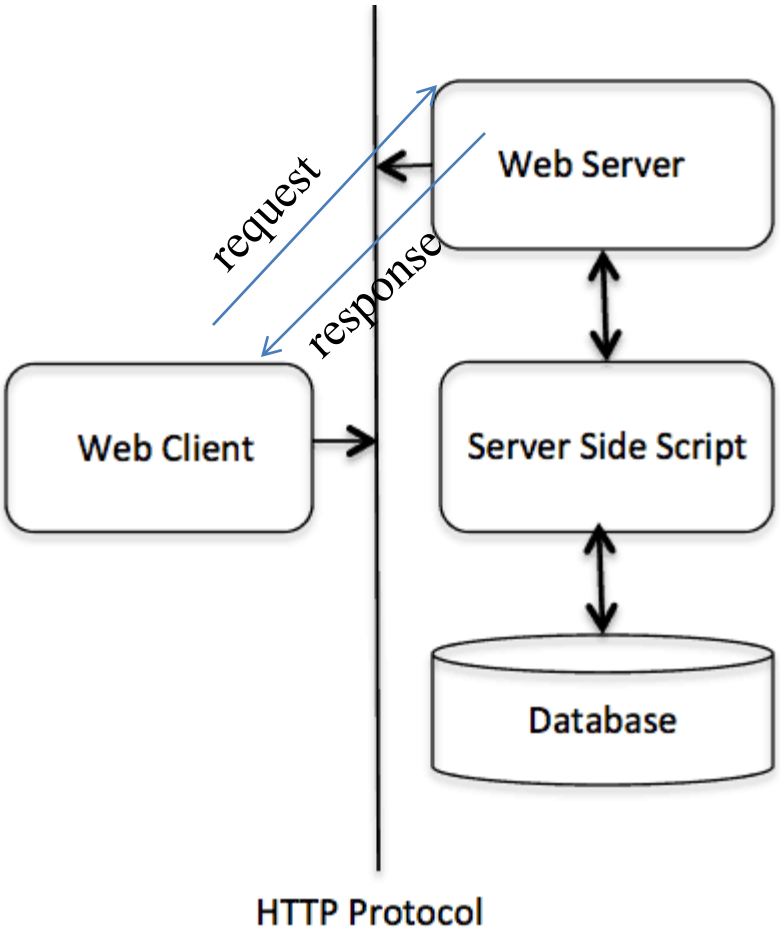
HTTP

HTTP протокол

- Протокол, позволяващ обмен на хипермедийна информация в Интернет.
- Използва се за реализирането на World Wide Web (WWW).
- Поддържа CGI (*Common Gateway Interface*), SSL (*Secure Socket Layer*) и виртуални домейни.
- Сървърният компонент е известен като “web сървър”, а клиентският – „browser”.
- За повишаване на сигурността се използва криптиране чрез протокол HTTPS.
- Използва се порт 80.

HTTP протокол

- Протокол от типа „заявка-отговор”
- Протоколът се реализира чрез съобщения в ASCII текст
- При изпращане на заявка се използва задължително FQDN



HTTP

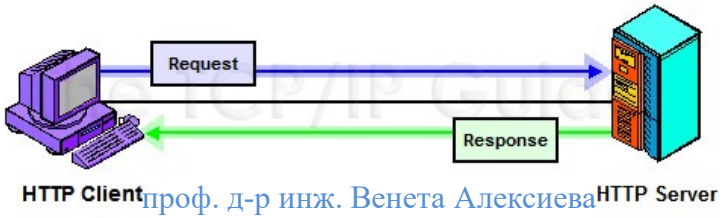
- Всеки хипермедиен обект се идентифицира посредством т.н. *Uniform Resource Locator (URL)*.
- Той се състои от три основни компонента: протокол, име на сървър и местоположение на ресурс:

http://www.nasa.gov/exploration/home/index.html

протокол | име на сървър | местоположение на ресурса

HTTP преди

- HTTP/1.0 всяка TCP конекция включва само един обмен.
- При HTTP/1.1 са възможни множество обмени.
- В зората на WWW хипертекстовите документи са били основно ТЕКСТОВИ документи.
- Една сесия е била достатъчна, тъй като целият ресурс е бил в един файл.
- През 1990 започва използването на хипермедийни документи, с включени графики и други файлове.



HTTP характеристики

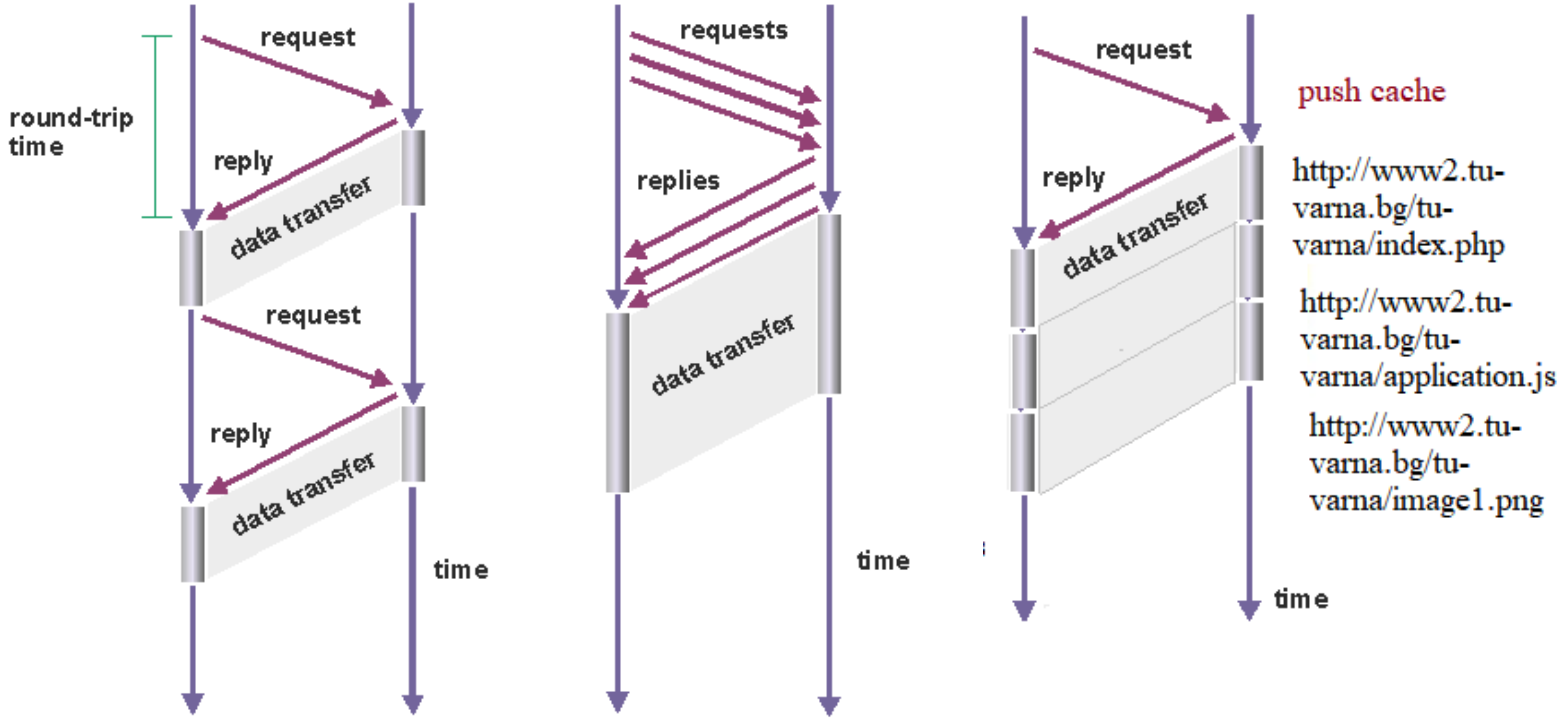
- Connectionless – доставя се ресурса и се прекъсва сесията
- Независим от преносната среда
- Stateless- не поддържа информация за състоянието, не помни взаимодействието с клиента
- Проблем, когато сесията трябва да бъде удължена – решава се със сесии и cookies
- HTTP1.0 – временни конекции и се затварят
- HTTP1.1 – постоянни конекции и не се затварят
- HTTP2.0 (от 2015г.) –мултиплексиране на заявки и отговори
- Няма механизъм за управление на диалога
- Управлението на потока от данни – от TCP

Transitory (временни) конекции

- Те са били единственият тип, поддържан от оригиналния HTTP/0.9.
- Неефикасни са когато клиентът трябва да направи множество заявки към един и същ сървър. Това е характерно за съвременните хипертекстови документи, съдържащи референции към изображения или медийни файлове.
- С временните конекции, всяка от тези отделни заявки от страна на клиента изисква да се установи отделна TCP конекция между клиента и сървъра.
- Всяка конекция изисква ресурси от страна на сървъра и заемане на мрежата.

Persistent (постоянни) конекции

- Те повишават ефективността на HTTP чрез премахване на повечето от разходите за комункация за HTTP/1.0.
- Те стават стандарт за комуникациите между клиентите и сървърите в Web.
- Създаване на постоянна конекция- клиентът е инициаторът на връзката с Web сървъра.



HTTP методи

- При отваряне на web страница, браузърът може да изпълни множество заявки към един или няколко сървъра за изтегляне на различните ресурси, описани в страницата.
- GET- изисква страница от web сървъра с даден URI (Uniform Resource Identifier)
- HEAD- като GET, но не връща body в отговора
- POST - изпращат се данни
- PUT- изпращат се (upload) ресурси към сървъра
- OPTIONS - определя опциите и/или изискванията към ресурсите на сървъра, без да предприема действие
- DELETE – изтрива посочения ресурс (index.html)
- CONNECT – за превключване между тунели с SSL
- TRACE – да изпрати заявка до дестинацията (сървър или първия прокси), която да отговори на клиента

SMTP/POP3/IMAP4

Електронната поща преди

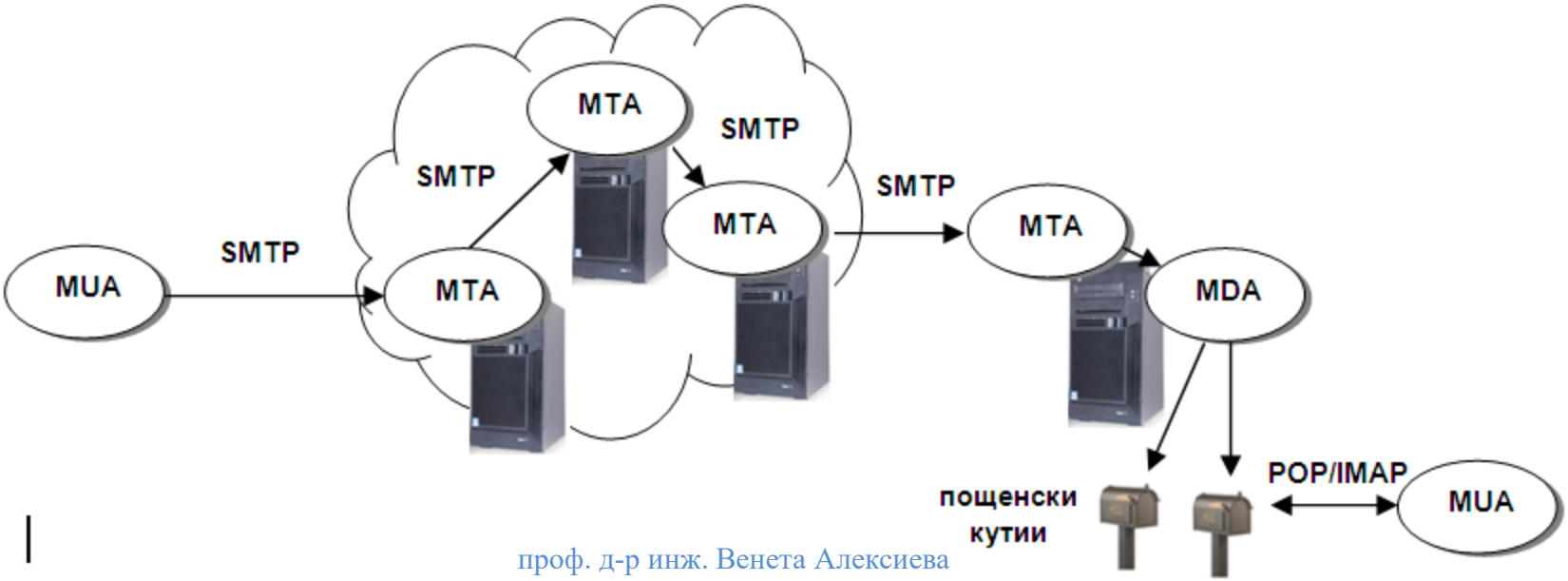
- Възниква 1965г за копиране съдържанието на файл от една машина на друга като го добавя в т.н. *пощенска кутия* (*mailbox*) на получателя.
- Позволява трансфер само на текст.
- 1971г. в ARPANET се създава 1-во приложение за електронна поща – знакът @ **отделя името на потребителя от името на машината**
- 1977г.- RFC733- за унифициране на е-пощата
- 1982г.- RFC822 – за унифициране имената на домейни
- 1982г.- RFC821(RFC5321 от 2008г.) – SMTP, който заменя FTP за обмен на е-поща
- 1996г.- RFC2045- RFC2049 - MIME (**Multipurpose Internet Mail Extensions**)

Електронната поща сега

- Електронната поща не може да функционира без DNS система.
- Всяко електронно съобщение включва заглавни части (хедъри) и тяло.
- Протоколите SMTP, POP и IMAP използват команди в ASCII текст за управление на обмена.
- Използва се порт 25.

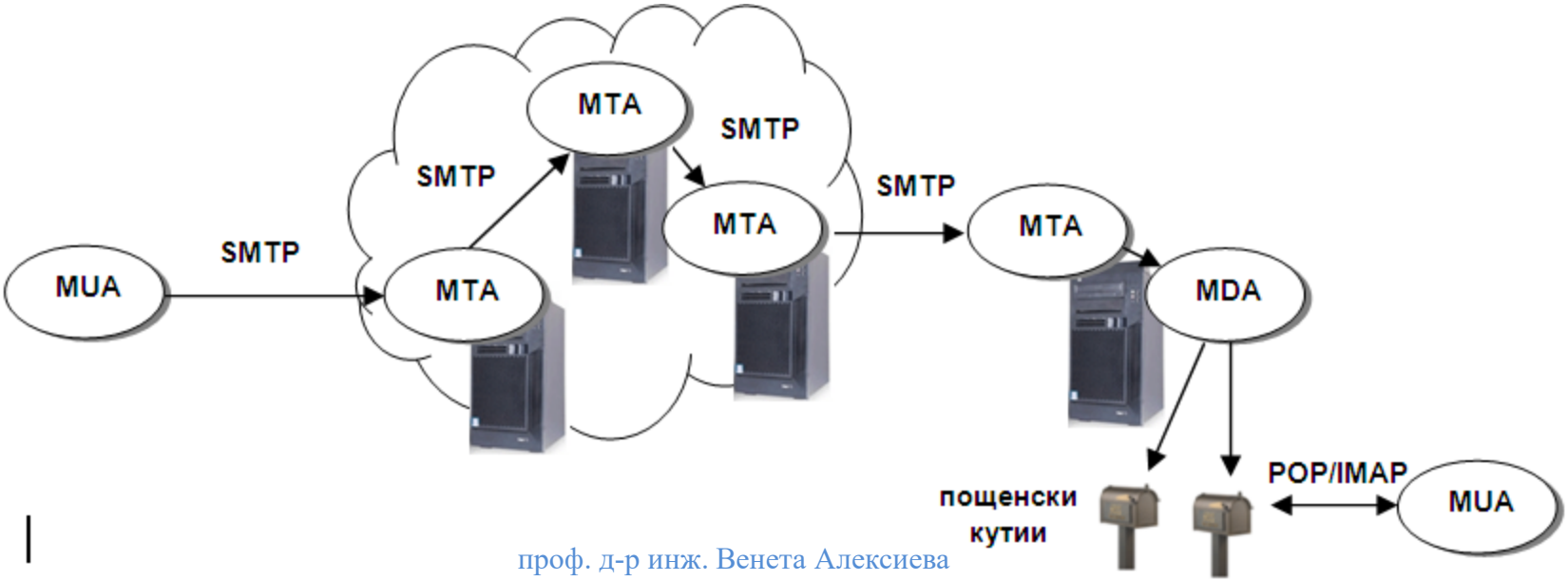
Основни компоненти

- **MUA** (*Mail User Agent*)
  - **При предаване-**формиране на e-mail съобщение и изпращането му към друга програма, която да осъществи неговото доставяне.
  - **При получаване -** прочитане на съобщенията от пощенската кутия на потребителя, класифициране и обработка.
- **Примери:** *mail* и *pine* под Linux, *Pegasus Mail* и *Outlook Express* под Windows.



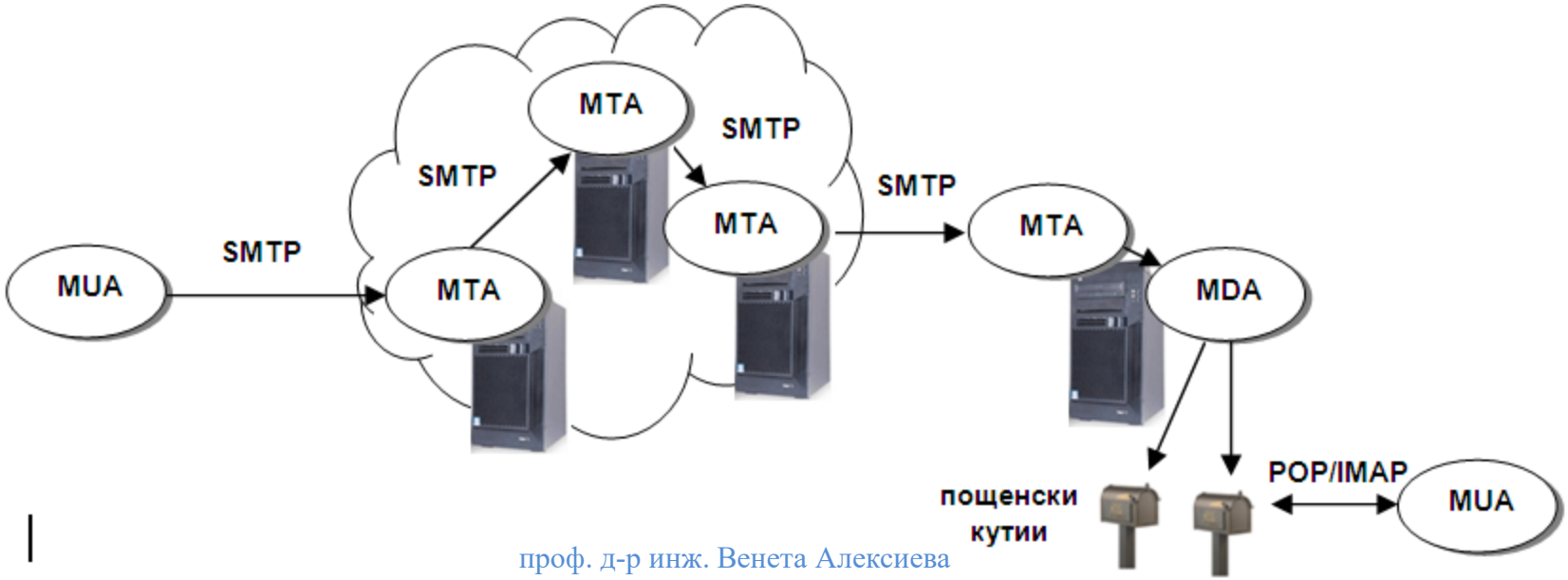
Основни компоненти

- **MTA** (*Mail Transport Agent*) е специализирана програма, осъществяваща транспортирането на електронната поща между машините и нейната доставка до получателя.
  - получава подготвените съобщения от MUA,
  - определя дестинацията,
  - препраща писмата (*relay*) през други междинни MTA до машината на получателя – т.н *пощенски сървър* (*Mail server*).
- **Протоколи**:
  - **UUCP** (*Unix to Unix Copy*) –за обмен на електронна поща между компютърни системи под Unix;
  - **X.400** – за електронна поща в системи, базирани на X.25 мрежови протокол;
  - **SMTP** (*Simple Mail Transport Protocol*) –за електронна поща в съвременния Интернет
- **Примери**: под Unix и Linux - *sendmail*, *smail*, *Postfix* и др.



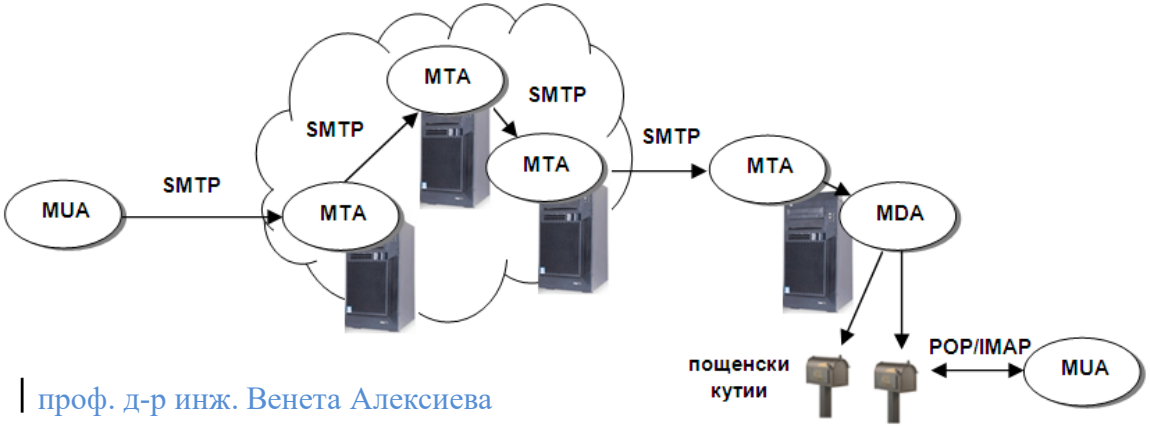
Основни компоненти

- **MDA** (*Mail Delivery Agent*) осигурява доставянето на полученото от пощенския сървър писмо в пощенската кутия на конкретния получател.
  - използване на псевдоними (*aliases*),
  - пренасочване (*forwarding*)
  - списъци с получатели на електронна поща (*mail lists*).
- **Примери:** под Linux се изпълнява от програмата *mail*.



Стъпки

- **Създаване на електронната поща**.
  - MUA (от компютъра на потребителя) формира съобщение.
- **Изпращане на съобщението**.
  - MUA изпраща готовото писмо на съответния MTA по SMTP (RFC2822) или по неговото разширение ESMTP.
- **Маршрутизиране на съобщението**.
  - Ако получателят е на същата машина, MTA извършва локално доставяне, с използване на услугите на локалния MDA.
  - Ако получателят е на отдалечена машина, MTA осигурява предаването на съобщението до дестинацията по SMTP или ESMTP.
- **Разпределяне на съобщенията**.
  - След получаване на електронната поща в крайния пощенски сървър, тя се обработва от съответния MDA и са записва в пощенската кутия на получателя.
- **Прочитане на съобщението**.
  - Потребителят прочита полученото писмо от своята пощенска кутия с **POP3** (*Post Office Protocol ver.3*), чрез който съдържанието на пощенската кутия на потребителя се прехвърля на локалния му компютър или **IMAP4** (*Internet Message Access Protocol ver.4*), чрез който потребителят може освен това и да класифицира, изтрива и поддържа своите електронни писма динамично на пощенския сървър.



Адреси при електронната поща

Адрес на получателя:

идентификатор_на_локалния_потребител @ описание_на_домейн

- ***идентификатор_на_локалния_потребител*** **(**изпращача или получателя)
  - потребителското име на съответната машина (*login name*),
  - Реалното име във формат ***ПървоИме.ПоследноИме***
  - псевдоним, който ще бъда преобразуван в потребителско име
  - списък от потребители.
- ***описание_на_домейн (***пощенския сървър на изпращача или на получателя)
  - това е пълното квалифицирано име на домейн (FQDN) във вида ***host.domain***.
  - Това обозначение не описва пътя до крайната машина.

Протокол SMTP

- Протоколът SMTP позволява клиентите да доставят електронната поща до съответните пощенски сървъри.
- Сървърите използват също SMTP за да препратят съобщенията между други пощенски сървъри до окончателното доставяне на писмото в пощенската кутия на получателя.
- Стартиране на SMTP сесия се извършва винаги от страна на MUA, докато транспортният агент очаква получаването на SMTP съобщения на порт 25.
- ESMTP (Extended Simple Mail Transport Protocol, RFC1869).- транспортирането на 8-битови ASCII символи, поддържането на MIME, индикация за максималната големина на съобщението и др.

Получаване на поща

3 базови модела:

- Off-line (POP3, IMAP4) – изтеглят се съобщенията на локалния компютър
- On-line (IMAP4) - съобщения се манипулират директно на пощенския сървър без необходимостта от тяхното изтегляне на локалния компютър.
- Disconnected use (IMAP4) – свързва се със сървъра, синхронизира съобщенията като изтегля в локалния кеш новите и приключва връзката

Въпроси ?

Благодаря за вниманието !