8. Компютърни мрежи и протоколи – OSI модел. Протоколи IPv4, IPv6, TCP, HTTP

Анотация: OSI модел – най-обща характеристика на нивата, съпоставане с модела TCP/IP. IPv4 адресация – класова и безкласова. Основни характеристики на протокол IPv6. TCP – процедура на трикратно договаряне. Хипертекстов протокол HTTP.

OSI модел най-обща характеристика на нивата

OSI (Open Systems Interconnection) моделът е теоритичен модел, описващ принципите на комуникация в мрежа и мрежовите протоколи. Основната градивна единица са слоеве, като всеки слой предоставя интерфейс и услуга към слоя над него.

1. **Физическият слой** е най-ниският слой на OSI модела. Отговаря за приемането и предаването на неструктурирани потоци от данни (битове) по физическия носител. Той отговарят за физическата връзка между върховете в дадена топология. Функциите на физическия слой включват:

   (а) Синхронизация на битовете – осигурява се чрез общ часовник, контролиращ изпращача и получателя;
   (б) Честота на изпращане на битове (bit rate control);
   (в) Определяне на мрежовата топология bus/star/mesh;
   (г) Определяне на режим на предаване на битовете – simplex/half-duplex/full-duplex;
   (д) Начин на установяване/прекъсване на връзката между устройствата;
   (е) Кодиране/декодиране на информация (в двоичен вид) към и от електрически сигнали.

   Обектите на този слой са хардуерни устройства – мрежови карти (NIC), модули, модеми, коаксиални кабели и други.

2. **Каналният слой** се грижи за безгрешното предаване на данни от едно устройство до друго, утилизирайки физическия слой. Може да бъде разделен на два подслоя: **Logical Link Control (LLC)** и **Media Access Control (MAC)**.
   Отговорност на DLL е да предава пакетите до правилните хостове използвайки техните MAC адреси, който се намират посредством Address Resolution Protocol (ARP). Когато пакет от мрежовия слой пристигне, той бива допълнително разбит на кадри (frames) в зависимост от размера на кадрите дефинирани от мрежовите карти (Network Interface Card (NIC)). Функциите на каналния слой включват:

   (а) Организира последователност на кадрите (фреймовете) – поставя специални битови последователности в началото и края на всеки кадър;
   (б) Физическо адресиране чрез добавяне на MAC адресите на получателя и/или изпращача в хедърите на всеки кадър;
   (в) Контрол над грешките чрез откриване и предаване отново на счупени и/или загубени кадри;
   (г) Контрол над потока от данни – честотата на получаване на устройствата може да бъде различна и заради това се налага координация над количеството от данни, което може да се предава за даден интервал от време;
   (д) Контрол на достъпа – MAC подслоя се използва, за да се определи кой има контрол над даден комуникационен канал в някой момент, когато той се използва между много устройства.

DLL обичайно се предоставя от NIC и някои драйвери на устройства. Имплементира се от Ethernet, switch-овете и ridge-овете.

3. **Мрежовият слой** отговаря за предаването на данни между хостове от различни мрежи, потенциално различаващи се по физическите и каналните си слоеве. Той се грижи за управлението на пакетите в мрежата. Като част от този процес е възможно данните да бъдат фрагментирани под формата на пакети (Protocol Data Unit (PDU)). Функциите на мрежовия слой включват:

   (а) Маршрутизация – намиране на най-кратък път от изпращач до получател;

(б) Логическа адресация – начин за идентификация на всяко устройство в мрежата. При протокола IP адресите на изпращача и получателя биват поставени в хедъра на пакета;
(в) Предотвратяване на натоварването на мрежата.

4. **Транспортният слой** се грижи за цялостта на предаване на съобщенията, пристигането им в точната последователност, потвърждаване за пристигане и проверка за загуби и дублиращи се съобщения.

   TCP (Transmission Control Protocol) е протокол ориентиран към връзката, което означава, че след като връзката е установена, данните могат да се предават в две посоки. TCP има вградени системи за проверка за грешки и за гарантиране, че данните ще бъдат доставени в реда, в който са изпратени. Протоколите от този слой разбиват съобщенията получени от сесийния слой на по-малки наречени сегменти и ги препращат към мрежовия слой, осигурявайки, че всички части са пристигнали цели и в правилна наредба при получателя. Ако сегмент не е достигнал до дестинацията си, то той бива изпратен отново. Това го прави идеалният протокол за прехвърляне на информация като неподвижни изображения, файлове с данни и уеб страници.
   Но докато TCP е инстинктивно надежден, неговите механизми за обратна връзка също водят до по-голямо натоварване, което води до по-голямо използване на наличната честотна лента в мрежата.

   UDP (User Datagram Protocol) е по-прост интернет протокол без връзка, при който не се изискват услуги за проверка на грешки и възстановяване. С UDP няма допълнителни разходи за отваряне на връзка, поддържане на връзка или прекратяване на връзка. Данните се изпращат непрекъснато до получателя, независимо дали той ги получава или не.
   Въпреки че UDP не е идеален за изпращане на имейл, разглеждане на уеб страница или изтегляне на файл, той до голяма степен се предпочита за комуникации в реално време като излъчване или многозадачно мрежово предаване.

   Функциите на транспортния слой включват:

   (а) Сегментация и повторно сглобяване – транспортния слой на изпращача получава съобщението от сесийния слой, разбива го на сегменти, поставяйки в хедъра на всеки – метаданни за реда на реасемблиране и ги подава на мрежовия слой. Транспортният слой на получателя се грижи за реасемблирането на сегментите на база техните хедъри;
   (б) Адресация на услуги (Service Point Addressing) – за да бъде доставено съобщението до правилния процес върху дестинацията, транспортния слой добавя порт на дестинацията в хедърите на всеки сегмент.

   Транспортният слой се имплементира като част от ОС, правеща системни извиквания към процесите.

5. **Сесийният слой** предоставя:

   (а) Създаване, поддържане и терминиране на сесии между две програми, които си комуникират;
   (б) Синхронизация – процесите могат да добавят контролни точки в данните чрез периодичното им запазване. Така след прекъсвания на предаването и/или грешки се избягва загуба или повреда на данните;
   (в) Осигурява различните режими на комуникация: еднопосочен диалог (simplex), двупосочен едновременен диалог (full-duplex), двупосочен алтернативен диалог (half-duplex);
   (г) Предоставя възможност за прекъсване на комуникацията и възстановяването ѝ от мястото на прекъсване;
   (д) Сигурност (например автентикация).

Обичайно се имплементира чрез сокети.

6. **Презентационният слой** извлича и манипулира данните от приложния слой за да станат годни за пренос по мрежата. Той предоставя:

   (а) Транслация – Конвертиране на данновия формат;
   (б) Криптиране/Декриптиране;
   (в) Компресия.

7. **Приложният слой** позволява на потребителските приложения да заявяват услуги или информация, а на сървър прионенията – да се регистрират и предоставят услуги в мрежата. Той представлява съвкупност от протоколи, които се имплементират директно от процесите. Примери са **HTTP** и **DNS**:

OSI model	
# Layer	**Protocols**
7. Application	DNS, FTP, HTTP, NFS, NTP, DHCP, SMTP, Telnet
6. Presentation	MIME, XDR, TLS/SSL, IMAP, FTP
5. Session	Named Pipes, SAP, PPTP, Sockets
4. Transport	TCP, UDP
3. Network	IPv3, IPv6, IPX, ICMP
2. Data Link Layer (DLL)	ATM, X.25, DSL, IEEE 802.11
1. Physical	IEEE 802.11, IEEE 1394, Bluetooth

Съпоставяне на OSI модела с TCP/IP
Когато започват да се изграждат реални мрежи, използвайки OSI модела и съществуващите протоколи, се вижда, че те не отговарят на изискваните спецификации за обслужване. OSI модела е просто „**наръчник**“ – той е разработен с цел да опише функциите на комуникационната система, като самата процедура по комуникация се разбива на малки и прости компоненти. За разлика от него, TCP/IP (Transmission Control Protocol/Internet Protocol) модела е разработен на базата на страндартни протоколи през 60-те години на миналия век. Той е компактна версия на OSI модела, като съдържа 4 слоя: приложение, транспортиране, маршрутизация и достъп до мрежата. Слоевете транспортиране и маршрутизация директно съответстват на слоевете транспортен и мрежови съответно от OSI модела. Слоят приложение събира слоевете приложен и презентационен съответно от OSI модела, а слоят достъп до мрежа събира първите два слоя от OSI модела (физически и канален).

Двата модела си приличат по това, че имат единен стек от независими протоколи и изпълняват подобни функции.

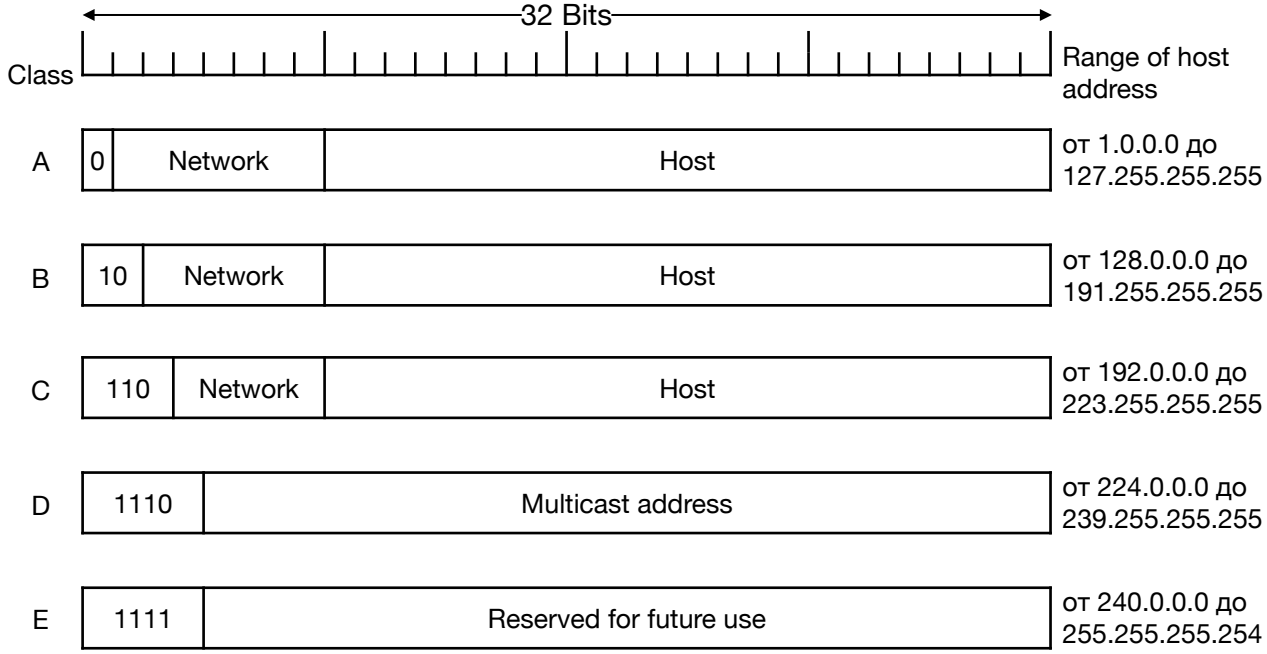
Разликите между двата модела са:
- TCP/IP е световен мрежов стандарт и на базата за него се изгражда концепцията за интернет. OSI модела не е приложим в това отношение;
- OSI модела прави ясно разграничение между основните си свойства, които са дефиниране на услуги, на интерфейси и на протоколи. TCP/IP модела, обаче не налага точно разграничение между тези три свойства;
- Протоколите в OSI са по-добре обособени и могат да бъдат заменяни по-лесно, поради това, че OSI е бил разработен преди да е създадена концепцията за протоколи (тоест е достатъчно общ). От друга страна TCP/IP модела е изграден на базата на протоколи, откъдето следва, че не могат да се заменят толкова лесно;
- Докато при OSI е нужно напасване на протоколите към самия модел, при TCP/IP не е нужно такова напасване, тъй като той самият е описание на вече съществуващи протоколи;

– На мрежово ниво TCP/IP е само connectionless, а OSI е connection oriented;
– На транспортно ниво OSI е само connection oriented, а TCP/IP е и connectionless чрез UDP и connection oriented чрез TCP.

IPv4 адресация – класова и безкласова. Основни характеристики на протокол IPv6
Адресите в IPv4 са 32-битови двоични числа, които се разбиват на 4 части, наричани октети. Един октет има 8 бита и се разделя от останалите октети с точка. Всеки един октет се записва като десетично число между 0 и 255. Мрежовите адреси може да разделим логически на две части: мрежови идентификатор (netid) и идентификатор на хоста (hostid). Ако двоичното представяне на идентификаторът на хоста се състои само от нули, то това е адресът на мрежата, към която хостът принадлежи. Ако пък двоичното представяне на hostid се състои само от единици, то това е адреса за едновременно предаване до всички машини в дадената мрежа. Например 178.255.255.255 адресира всички хостове в мрежата 178.2.0.0.

Някои резервирани адреси:
– 0.0.0.0 – маршрут по подразбиране, който се използва при маршрутизация на IP дейтаграмите
– 127.0.0.0 – резервиран за локален IP трафик; присвоява се интерфейс за обратна връзка към хоста – loopback interface
– 127.0.0.1 (както и 127.x.x.x) – използва се за служебен адрес на самия хост (localhost address)

В IPv4 адресите се делят на следните 5 класа:



Фиг. 1. Класове от IP адреси

- **Клас A** – фиксираният префикс е **0**, т.е. обхвата на първия октет е **0000 0001 – 0111 111**, т.е. е от **1 – 127**.
  Subnet маската на клас A по подразбиране е **255.0.0.0**.
  Клас A включва адресите само от **1.0.0.0** до **126.255.255.255**, като **127.x.x.x** е резервиран за loopback адреса, а **0.0.0.0** е не-рутируем мета-адрес използван по различен начин в зависимост от контекста, като обичайно е за незнаен или невалиден адрес.
  Следователно num_networks = 2^7 - 2 = 126 и num_hosts = 2^{24} - 2.
- **Клас B** – фиксираният префикс е **10**, т.е. обхвата на първия октет е **1000 0000 – 1011 1111**, т.е. е от **128 - 191**.
  Subnet маската на клас B по подразбиране е **255.255.0.0**.

Клас B вйлючва адресите само от **128.0.0.0** до **191.255.255.255**.
Следователно num_networks = 2^{14} и num_hosts = 2^{16}- 2.

- **Клас C** – фиксираният префикс е **110**, т.е. обхвата на първия октет е **1100 000 – 1101 1111**, т.е. е от **192 – 223**.
  Subnet маската на клас C по подразбиране е **255.255.255.0**.
  Клас C включва адреси само от **192.0.0.0** до **233.255.255.255**.
  Следователно num_networks = 2^{21} и num_hosts = 2^{8}- 2.
- **Клас D** – фиксираният префикс е **1110**, т.е. обхвата на първия октет е **1110 0000 – 1110 1111**, т.е. е от **224 – 239**.
  Клас D е резервиран за multicasting и няма subnet маска.
  Следователно щом данните при multicasting не са насочени към конкретен хост няма нужда от извличане на host адреси.
- **Клас E** – фиксираният префикс е **1111** и е резервиран за експериментални дейности в бъдещето.
  Обхватът на клас E е **240.0.0.0 – 255.255.255.254**. Той не разполага със subnet маска.

Така наречените частни интернет мрежи (private networks) са резервирани за използване от различни организации за реализиране на връзка между компютрите само в рамките на съответната организация. Такива мрежи са: 10.x.x.x от клас A, 16 мрежи от 172.16.x.x до 172.31.x.x от клас B и 256 мрежи 192.168.x.x от клас C. Адресите от тези мрежи не се маршрутизират в глобалния Интернет. Най-честото им приложение е при firewall-и.

Големият недостатък на IPv4 е, че половината адреси са от клас A и се разпределят само измежду 127 автономни системи, въпреки че всяка от тях може да съдържа милиони хостове. Всяка мрежа трябва да има уникален номер и всички хостове в дадена мрежа трябва да имат един и същ номер на мрежата. Това води до проблеми при нарастване на броя мрежи.

С разрастването на Интернет, най-бързо са изчерпани свободните IP адреси на мрежи от клас B, което налага на организации, притежаващи голям брой компютри, да се дават две или повече мрежи от клас C. Това води до увеличаване на размерите на глобалните таблици: от една страна, защото в маршрутизиращите таблици има по няколко записа, които водят до общ маршрутизатор на организацията, а от друга – заради произволното раздаване на номера на мрежи, което налага маршрутизаторите да пазят запис за всяка мрежа, без възможност за агрегиране. За намаляване на този обем се въвежда безкласова адресация и маршрутизация – CIDR (Classless Inter-Domain Routing). CIDR записът представя всеки адрес като 32-битово число, последвано от наклонена черта"/" и броя единици в двоичния запис на subnet маската. Например, 192.0.2.96/28 означава IP address, в който първите 28 бита са за netid, а subnet маската е 255.255.255.240.

Тъй като IPv4 адресите са вече изчерпани, преходът към IPv6 е неизбежен. Този преход, обаче, се извърпва бавно, поради това, че IPv6 не поддържа обратна съвместимост и са необходими промени в мрежовите устройства и услуги. IPv6 решава следните задачи:

– Възможност за достъп до глобалната мрежа на милиарди хостове, даже при нерационално използване на адресното пространство;
– Съкращаване на размера на маршрутните таблици;
– Опростяване на протокола с цел улесняване на обработването на пакетите при маршрутизацията им;
– Повишаване на нивото на безопасност на протокола.

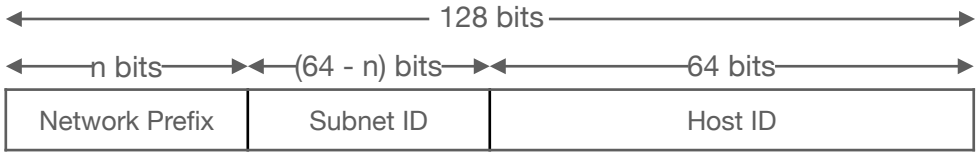
Основни характеристики на протокол IPv6

Протоколът IPv6 поддържа основните функционалнотси на IPv4, но с основната разлика, че е пълен редизай на IPv4 и не е обратно съвместим.
Някой от основните характеристики на IPv6 са:

- По-голямо адресно пространство. При IPv6 адресите са 128-битови, което означава, че са 4 пъти по-широки от при IPv4, следователно броят им е 2^{96}. Това означава, че

IPv6 поддържа около $3,4 \times 10^{38}$ различни адреса. Това са 1564 адреса на квадратен метър от земята и следователно всяко устройство може да разполага със свой уникален IPv6 адрес. По този начин се реашава проблема с изчерпването на всички IP адреси;
- Опростен хедър. Много от ненужните полета при IPv4 са преместени в края на хедъра на IPv6, като те са опционални. Следователно хедъра е разширяем. Така IPv6 хедърите може да са около двойно по-големи от IPv4 хедърите, при положение, че адресите са 4 пъти по-дълги;
- Комуникация от край-до-край. Тъй като всеки хост има свой уникален IP адрес, той може да траверсира мрежата до всеки друг хост без нужда от NAT (Network Address Translation);
- Автоматична конфигурация. IPv6 поддържа stateful и stateless автоматична конфигурация на устройствата. Така неналичието на DHCP (Dynamic Host Configuration Protocol) не спира комуникацията между мрежите.
- По-бързо рутиране и предаване – заради съкратения хедър;
- IPSec – по-сигурен е от IPv4;
- Няма broadcast – не са специфицирани broadcast адреси в IPv6, като се използва multicast за комуникация с много адреси;
- Anycast – много хостове могат да получат един и същ IPv6 адрес, като пакетите се рутират до най-близкия такъв;
- Мобилност. Позволява на хостовете да се преместват географски, но да задържат своя IP адрес. Това се постига чрез ползване на автоматичната конфигурация и допълнителни хедъри;
- Плавен преход от IPv4 до IPv6.

Адресите в IPv6 са 128-битови, като те се разбиват на 8 групи с по 4 шестнадесетични числа, разделени с двуеточие (Например 2001:0db8:9095:02e5:0216:cbff:feb2:7474).



Фиг. 2. IPv6 адрес

Един IPv6 адрес се състои от:
– Мрежови префикс (network prefix) – идентифицира дадена мрежа или специален адрес;
– Идентификаторът на подмрежата (subnet ID) – връзка вътре в мрежов обект. Присвоява се от администратора на обекта. Определя на кой мрежов сегмент принадлежи даден хост;
– Host ID – идентифицира конкретен възел в мрежата – конкретен негов интерфейс.
Мрежовият префикс е аналогичен на означението с "/" в IPv4, например мрежата 2001:0db8:9095:02e5:0216:cbff:feb2:7474/32 (в този случай 2001:0db8 е мрежовият префикс, 9095:02e5 е идентификаторът на подмрежата, а останалите групи образуват Host ID).

За да се улесни записването на адреси, съдържащи нули, те се компресират по определени правила. "::" символът се появява, когато адресът съдържа една или повече 16-битови групи от нули. Този символ се използва за компресиране на водещи или завършващи нули и може да се появи най-много веднъж в адреса.

Типовете IPv6 адреси са следните:

– Неопределени – ::/128 (00..0 в битово представяне, тоест 128 битове 0);
– Loopback – ::1/128 (00...1 в битово представяне);
– Multicast – FF00::/8 (11111111 в битово представяне);
– Link-Local Unicast – FE80::/10 (1111111010 в битово представяне);
– Global Unicast – всички останали

Забелязва се, че липсват broadcast адреси и това е така, тъй като те не са дефинирани в IPv6. Multicast адресирането поема и тяхната функция в IPv6.

TCP – процедура на трикратно договаряне
TCP (Transmission Control Protocol) е протокол от транспортния слой, като той цели да транспортира по надежден начин потока от байтове получен от по-горните слоеве. Това се постига чрез записването на поредни номера в хедърите на сегментите. Така дори и да има повторени, загубени или разбъркани пакети, данните биват сглобявани по коректен начин при получателя.

Други характеристики на TCP са:
- **full-duplex** комуникацията е разрешена;
- **управление на потока (flow-control)** – синхронизация на скоростта на предаване на данните между изпращача и получателя. Постига се чрез полето **window size**, което определя какво количество данни да бъдат изпратени от изпращача преди да бъде получено потвърждение. Първоначалният window size се установява на база **MSS (Maximum Segment Size)**;
- statefulness – обмена на данни се случва след установяване на връзка.

Връзка се осъществява чрез процедурата **трикратно договаряне (three-way handshake)**:

Първоначално отваряне на връзката (Connection Establishment Protocol): Необходимо е всеки един от двата хоста да изпрати на другия началния номер (initial sequence number) на байтовата последователност, която ще изпраща, и съответно да получи насрещното потвърждение за получаването на този номер. Процедурата, описана по-долу, използва SYN (synchronization) сегмент, който се използва за изпращане на началния номер за синхронизация на номерацията на сегментите. Процедурата е следната:

1. **SYN** – Клиентът изпраща сегмент в вдигнат **SYN (Synchronize Sequence Number)** бит, съдържащ **пореден номер (sequence number)** до сървъра, от който започват сегментите, които той ще изпраща;

2. **SYN + ACK** – Сървърът отговаря със сегмент с вдигнати **SYN** и **ACK (Acknowledgement)** битове. ACK се използва за потвърждение на първия сегмент изпратен от клиента, а **SYN** за да се укаже с какъв **пореден номер** ще започнат сегментите изпратени от получателя;

3. **ACK** – Клиентът потвърждава, че е получил отговора на сървъра, след което се установява full-duplex връзка и започва преноса на данни.

Хипертекстов протокол HTTP
HTTP (Hypertext Transfer Protocol) е протокол на приложния слой. Той представлява прост текстов протокол, който се използва от услугата WWW за осигуряване на достъп до практически всякакъв вид данни, наричани събирателно ресурси.
При HTTP протокола има понятия като клиент (обикновено това са уеб браузърите) и сървър (това са уеб сървъри). HTTP най-често се използва редом с TCP/IP, но практически може да работи върху всякакъв протокол, който предоставя надежден транспорт. Обикновено HTTP протоклът работи върху стандартен TCP сокет, отворен от клиента към сървъра. Стандартният порт за HTTP е 80, но може да се използва и всеки друг TCP порт.

Комуникацията по HTTP се състои от заявка (request) – съобщение от клиента към сървъра, и отговор (response) – отговор на сървъра на съобщението от клиента. HTTP заявките имат 3 основни елемента: метод на достъп, Request URI и header полета.
Методът описва вида на HTTP заявката, изпратен от клиента. HTTP дефинира 9 метода, които могат да се използват за извърпване на действие от конкретен тип върху заявения ресурс:

- **GET** – Заявява репрезентация на конкретен ресурс. Използва се само за извличане на данни;
- **HEAD** – Същия метод като GET, но без да изисква изпращане на съдържание (response body), т.е. samo хедър (response header);

- **POST** – На база на request body променя заявения ресурс ако съществува или го създава в противен случай;
- **PUT** – Заменя всяко представяне на указания ресурс с това, което е в request body;
- **PATCH** – Частично модифицира заявения ресурс на база на request body;
- **DELETE** – Изтрива указания ресурс;
- **OPTIONS** – Връща поддържаните HTTP методи за заявения ресурс;
- **TRACE** – Прави тест на пътя към заявения ресурс, т.е. дали сървърът е получил нашето съобщение.

Наименованията на методите носят само семантичност на заявките.

Request URI определя ресурса, над който ще оперира заявката. Могат да се използват два вида идентификатори: URI идентификатор или релативен път спрямо главната директория на уеб сървъра. URI (Uniform Resource Identifier) е идентификатор на ресурс, определен или по местоположение чрез URL (Uniform Resource Locator, например http://www.example.com/folder/page.html), или по име чрез URN (Uniform Resource Name). Релативният път спрямо главната директория на уеб сървъра задава местоположението на ресурс в рамките на текущия уеб сървър. Това е частта от URL-а, която стои след името на хоста (сървъра) в URL идентификатора, например /folder/page.html.