ДЕЛЯН ГЕНКОВ

# Основи на компютърните мрежи

# ОСНОВИ НА КОМПЮТЪРНИТЕ МРЕЖИ



Българска

Първо издание

Авторът е водещ преподавател по дисциплините „Компютърни мрежи“, „Интернет технологии“, „Мрежова сигурност“ и „Бази данни“ в Технически университет – Габрово, катедра „Компютърни системи и технологии“. От 1999 г. е основател, главен контакт и инструктор, обучаващ инструктори и студенти в първата в България Cisco мрежова академия. Той има над 20 годишен практически опит в областта на компютърните мрежи и притежава редица международно признати сертификати, между които CCNP (Cisco Certified Networking Professional), CCNA R&S (Cisco Certified Networking Associated Routing and Switching), CCNA Security (Cisco Certified Networking Associated Security), CPTE (Certified Penetration Test Engineer), Cisco Networking Academy Instructor Trainer, Certified Cisco Systems Instructor, Cisco Certified Academy Instructor. За дейността си в мрежовата академия е отличен от Cisco Systems с наградата Instructor Excellence Expert.

ISBN 978-619-7071-61-0

Тази книга представлява учебник по дисциплината “Компютърни мрежи” и е предназначен за студентите от специалност “Компютърни системи и технологии” – ОКС „бакалавър” и ОКС „магистър” подготвителна степен от Технически университет - Габрово. Материалът е съобразен със учебната програма по тази дисциплина. Учебникът може да бъде използван и от студенти от други специалности, изучаващи подобни дисциплини или занимаващи се с проектиране, конфигуриране и поддръжка на компютърни мрежи.

С благодарност ще бъдат приети всички добронамерени бележки и препоръки, отправени към автора с цел подобряване на материала и отстраняване на евентуални грешки и пропуски.

Технически университет - Габрово,

ул. Х. Димитър №4.

Габрово, юни, 2014 Авторът

## Съдържание

# 1. Категоризиране на мрежите

## 1.1 Въведение в мрежите - понятия, определения.

В съвременния свят всички сме постоянно свързани с други хора, благодарение на мрежите. Използваме телефонните мрежи, за да говорим с познати и бизнес кореспонденти, гледаме телевизия благодарение на телевизионната мрежа, общуваме с приятели и партньори в социалните мрежи и използваме световната компютърна мрежа Интернет за всякакви разнообразни задачи.

Фокусът на тази книга е върху компютърните мрежи, които практически са в основата и на голяма част от останали видове.

Компютърната мрежа представлява съвкупност от две или повече компютърни системи, свързани заедно за обмен на информация или за изпълнение на някаква обща работа. Различните видове компютърни мрежи имат различни цели, размери и принципи на изграждане. Освен глобалната мрежа Интернет, към която ежедневно свързваме своите компютри, телефони, таблети и други устройства, за да черпим информация, да общуваме с останалите, да работим и да се забавляваме, компютърни мрежи има и на други места, за които вероятно не се замисляме. Например отделните важни възли на автомобила ни се управляват от специализирани компютри, които също са свързани в компютърна мрежа, за да си обменят информация. Банкоматите и POS-устройствата, от които теглим банкноти или плащаме сметки също работят благодарение на компютърната мрежа, която ги свързва с устройствата на банката.

Една от основните характеристики на една компютърна мрежа е нейната скорост на предаване на данни. Скоростта е важна, защото различните приложения имат различни изисквания за скорост. Например, скоростта необходима за гледане на един телевизионен канал със стандартна разрешаваща способност е приблизително равна на скоростта, необходима за провеждане на десет телефонни разговора едновременно.

Скоростите на предаване на информация в компютърните мрежи се измерват с няколко мерни единици. Първите две са единици за количество информация, а следващите за скорост.

- бит (bit) е основна единица информация в компютрите, представляваща най-малката информация, която може да се съхранява или предава. Един бит може да има стойност 0 или 1.
- Байт (byte) е съвкупност от осем бита. Това е единица информация, която може да представи една буква в писмо или документ.
- Бит в секунда (bps) – мярка за скорост на предаване на информация, при която за една секунда се предава един бит, т.е. за предаване на една буква са необходими осем секунди.

- Байт в секунда (Bps) - мярка за скорост на предаване на информация, при която за една секунда се предава един байт, което в повечето случаи означава една буква.

Тези мерни единици определят доста ниски скорости, в сравнение с типичните за съвременния мрежов свят, затова обикновено се използват техни производни:

- Килобит в секунда (kbps) – един килобит е равен на 1024 бита, т.е. при един килобит в секунда обикновено за една секунда се предават около 1024 / 8 или 128 букви.
- Килобайт в секунда (kBps) – 1024 байта или букви в секунда.
- Мегабит в секунда (Mbps) – 1024 килобита в секунда или 1048576 бита в секунда е основната мярка за скоростна свързаност към Интернет в съвременния свят.
- Мегабайт в секунда (MBps) – 1024 килобайта в секунда или малко над един милион букви в секунда.
- Гигабит в секунда (Gbps) – 1024 мегабита в секунда или 1073741824 (над един милиард) бита в секунда.

Често потребителите правят две основни грешки. Първата е свързана с навика от останалите мерни единици, като метър или грам, че представката „кило” означава 1000, „мега” – един милион и т.н. В компютърните мерни единици както се вижда по-горе увеличаването за всяка следваща представка е умножение по 1024, което може да дава известна разлика. Втората често срещана грешка е бъркането на битове и байтове. Когато потребител се абонира за достъп до Интернет, доставчикът на услугата обикновено определя скоростта на достъп в мегабитове в секунда. Когато се тегли информация от мрежата обикновено програмите показват текущата скорост в мегабайтове в секунда. Тогава потребителят често остава с погрешно впечатление, че при абонамент за достъп със скорост 40 мегабита в секунда той може да използва само около 5, т.е. че доставчикът не изпълнява задълженията си. Проблемът често се крие в различните мерни единици – 5 мегабайта в секунда по 8 бита в един байт е 40 мегабита в секунда.

## 1.2 Компоненти на компютърната мрежа

Различните компютърни мрежи могат да имат различни компоненти, но като общо описание могат да бъдат отделени следните:

### 1.2.1 Компютърни системи.

Компютърните системи са най-важната част от мрежите, защото всъщност мрежата е създадена, за да свързва тези системи. Компютърни системи могат да бъдат настолни или преносими компютри, таблети, смартфони или специализирани устройства. Понеже обикновено в една мрежова връзка се предоставят услуги и

ресурси, то в зависимост от това кой предоставя услугата и кой я използва, компютрите в мрежата се делят на:

- Клиенти – това са системите, използващи ресурси и услуги в компютърната мрежа;
- Сървъри – това са системите, предоставящи ресурси и услуги.

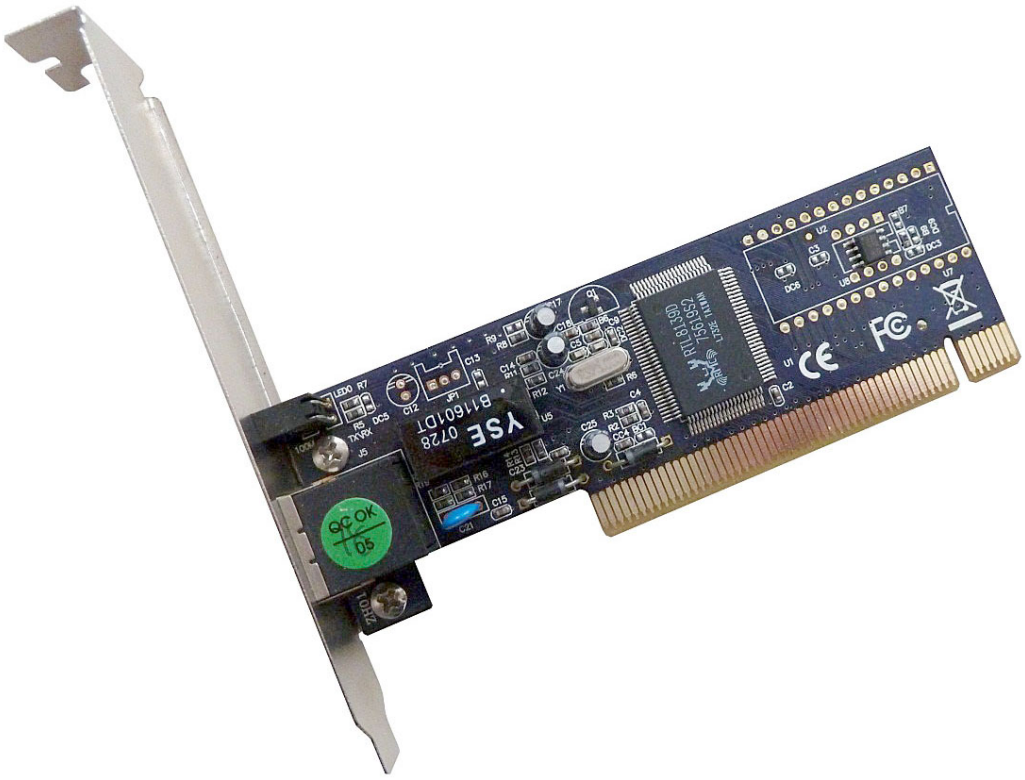
Възможно е един и същ компютър да играе роля и на сървър и на клиент – т.е. едновременно да предоставя услуга и да използва друга или същата услуга от друг компютър в мрежата.

### 1.2.2 Периферни устройства

Периферните устройства са принтери, скенери, уеб-камери, дискове за съхранение на информация и други подобни, които другите компютри използват чрез мрежата. Те могат да бъдат свързани към някой от компютрите в мрежата, който да ги споделя към останалите като свой ресурс или да си имат специализиран порт за мрежова връзка и самостоятелно да се свързват към мрежата.

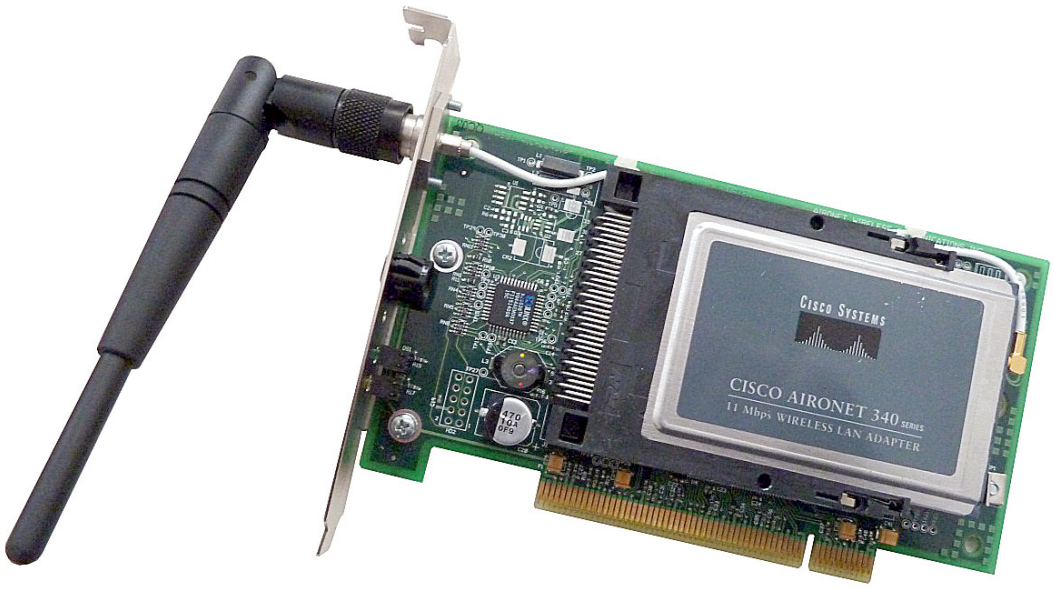
### 1.2.3 Мрежови контролери

За да може един компютър да бъде свързан в мрежа, той трябва да има мрежов контролер – устройството, което осъществява физическата връзка на компютъра с мрежата.



**Фигура 1.1. Мрежов контролер за жична (Ethernet) мрежа**

Мрежовите контролери могат да бъдат отделни платки, които се поставят на слот в компютъра и могат да се сменят при нужда или да бъдат вградени в дънната платка на компютъра. Съществуват различни технологии за компютърни мрежи и всеки контролер е специализиран за конкретната технология. На фигура 1.1 е показан мрежов контролер за жична мрежа, а на фигура 1.2 - за безжична мрежа.




**Фигура 1.2. Мрежов контролер за безжична (Wi-Fi) мрежа**

### 1.2.4 Преносна среда

Преносната среда е свързващото звено между устройствата в мрежата. Тя може да бъде във вид на специален кабел или да се използват радиовълни за предаване на информация в ефира. В съвременните компютърни мрежи се използват главно три вида преносни среди:

- Метални кабели – най-често изработени от медни проводници, по който сигналите се предават във вид на електрически импулси. Най-разпространените метални кабели са усуканите двойки проводници.
- Оптични влакна – проводници с прозрачна стъклена сърцевина, по която информацията се предава във вид на светлина. Това са най-перспективните преносни среди, с тях се постигат най-големи разстояния и най-големи скорости на предаване. Все още комуникацията по оптично влакно е по-скъпа в сравнение с останалите среди.
- Радиоефир – безжичните мрежи са доста разпространени и тяхното развитие продължава бързо, защото те са лесен, удобен и евтин начин за връзка между компютри, сгради, етажи без да се налага полагането на някакъв проводник. При тях информацията се пренася във вид на радиовълни и се предава по ефира, подобно на радио или телевизионен сигнал.

### 1.2.5 Мрежови устройства

Мрежовите устройства осъществяват връзката между средите на отделните компютри в обща мрежа или между вече изградената мрежа и друга мрежа – например Интернет. Примери за такива устройства са комутаторите (switch) за връзка между два или повече компютъра в локална мрежа и маршрутизаторите (router) за връзка на домашната мрежа с Интернет. Примерен домашен маршрутизатор е показан на фигура 1.3.


**Фигура 1.3. Маршрутизатор за малка мрежа.**

## 1.3 Видове компютърни мрежи.

Компютърните мрежи могат да се категоризират по много признаци. Най-често прилаганата категоризация е според техния размер.

### 1.3.1 Категоризация според размера на мрежата.

Обикновено размерът на мрежата се определя от максималното разстояние, което може да раздели компютърните системи в мрежата. Според този критерий, компютърните мрежи се делят на:

- Персонални мрежи (Personal Area Network, PAN) – това е компютърна мрежа свързваща личните устройства на един човек – смартфон, таблет, лаптоп,

персонален компютър. Обикновено целта на такава мрежа е обмен на файлове между отделните системи, например прехвърляне на снимки и клипове от телефона към компютъра. Типичен представител на тези мрежи е стандартът Bluetooth, вграден в повечето преносими устройства като основен начин за свързване. Обхватът на такива мрежи е от няколко метра до няколко десетки метра.

- Локални компютърни мрежи (Local Area Network, LAN) – Това е мрежа, свързваща компютърните системи, намиращи се в една стая, на един етаж, в една сграда или в група сгради на близко разстояние. Такива са повечето мрежи на учреждения, университети, фирми, хотели и други подобни организации, разположени в рамките на едно населено място. Максималното разстояние между системите в една локална мрежа зависи от използваната технология за връзка, но обикновено е между няколко десетки метра и няколко километра. Съществува специално понятие за малките локални мрежи, свързващи няколко компютърни системи в дома или в малък офис – те се наричат SOHO (Small-office Home-office) мрежи. Най-разпространената технология за изграждане на локални мрежи е етернет (Ethernet), разгледана по-назад. Напоследък голямо разпространение в локалните мрежи имат и безжичните мрежови технологии, обикновено наричани Wi-Fi.

- Квартални или градски мрежи (Metropolitan Area Networks, MAN) – компютърна мрежа, свързваща компютри в рамките на един или няколко квартала или на цял град. Обикновено покриваното разстояние е от няколко километра до няколко десетки километра. Въпреки че в България често такива инсталации се изграждат, използвайки технологията за локални мрежи Ethernet, технологиите за изграждане на градски мрежи са различни. Те включват достъп до Интернет през кабелната телевизионна мрежа, безжичен достъп с широк обхват, познат като Wi-Max, предаване на данни по телефонните кабели – DSL и някои по-съвременни методи, използващи за пренос оптични влакна – Metro Ethernet.

- Глобални мрежи (Wide Area Networks, WAN) – това са мрежи, покриващи няколко съседни града, цяла страна, няколко страни, цял континент или дори няколко континента, разполагащи се на разстояние от няколко десетки километра до десетки хиляди километри, колкото е обиколката на цялата земя. Обикновено тези мрежи не се изграждат за нуждите само на една организация, поради доста високата цена. Когато една организация се нуждае от връзка на своите офиси в различни страни или континенти, тя наема линии за връзка от телекомуникационните оператори. Така по техните трасета вървят едновременно данните на много организации. Технологии за достъп до глобални мрежи са наети линии, Frame Relay, SDH и други.

- Интернет (Internet). Това е световната компютърна мрежа, свързваща милиарди компютри в обща огромна мрежа. Тя е изградена от различни компютърни мрежи, използващи различни технологии. Самият термин „интернет" означава връзка между две и повече мрежи и такъв е смисълът, когато думата е изписана с малка буква. Когато се говори за световната мрежа е прието нейното име да се пише с главна буква – Интернет или Internet.

Примери за различни видове мрежи са представени на фигура 1.4.

| Разстояние между системите | Компютрите се намират в | Пример |
|---|---|---|
| 1 m – 10 m | Квадратен метър | Лична мрежа - PAN |
| 10 m | Стая | Локална мрежа - LAN |
| 100 m | Сграда | |
| 1 km | Университет | |
| 10 km | Град | Градска мрежа - MAN |
| 100 km | Държава | Глобална мрежа - WAN |
| 1000 km | Континент | |
| 10 000 km | Планета | Интернет |

**Фигура 1.4. Видове мрежи според размера.**

### 1.3.2 Категоризация според ролите на отделните компютри

- Равноправни мрежи (peer-to-peer)

При равноправните мрежи всеки компютър може да дава услуги на останалите в мрежата, както и да използва услугите, предоставяни от другите компютри. Равноправните мрежи са евтини за изграждане и са предназначени за малък брой компютри (според Microsoft – до 10). При тях обикновено на всеки компютър е инсталирана потребителска операционна система, например Windows 7 или Ubuntu Linux и на него може да работи потребител и да изпълнява програми. Всеки потребител сам решава какви ресурси на своя компютър (дисково пространство, принтери) ще позволи за използване от

останалите компютри в мрежата и какви права ще даде на другите потребители. Администрацията на този вид мрежи е разпределена – имената на потребителите, техните пароли и права се назначават отделно на всеки компютър и един потребител може да има различни пароли и права на различните компютри. Равноправните мрежи са подходящи за домашна обстановка, където всеки може да сподели снимки, видео и други документи с останалите и всеки да печата на принтера, свързан към единия компютър.

- Сървърно базирани мрежи

При сървърно базираните мрежи има един или повече специализирани компютри, наречени сървъри с инсталирана мрежова операционна система, която поддържа функциите на мрежата. Ако сървърът не работи, обикновено няма услуги, които да са достъпни през мрежата за компютрите на потребителите или мрежата не работи. Затова често сървърите се намират в специализирано помещение, без достъп на неоторизирани лица. Обикновено на сървърите не работят потребители, а те са отделени специално за да поддържат функционирането на мрежата, което оскъпява този вид мрежи, в сравнение с предишните. Понякога мрежовите операционни системи за сървъри могат да струват доста – от порядъка на хиляди лева. При сървърно базираните мрежи администрацията е централизирана – всички настройки, имена на потребители, пароли, права и други параметри се създават и променят еднократно само на сървъра и важат за цялата мрежа, което повишава и сигурността на тези решения. Сървърно базираните мрежи са типични за средни и големи офиси и учреждения, учебни заведения, университети и други организации с повече компютри и с нужда от по-голяма сигурност и стабилност на мрежата.

### 1.3.3 Категоризация според предоставяните услуги

- Мрежи за достъп до Интернет

Тези мрежи обикновено се изграждат от организации, наречени Интернет доставчици (Internet Service Providers, ISP) и са създадени с цел да предоставят на потребителите достъп до ресурсите на световната мрежа – Интернет. Често в такива мрежи доставчикът на услуги не предоставя допълнителни свои ресурси на потребителите (например сървъри за файлове или мрежови игри), а само им позволява те да достъпват чужди такива сървъри, намиращи се някъде в Интернет. Понякога потребителите на един и същ доставчик на Интернет услуги не могат да споделят помежду си услуги, например да си обменят файлове, въпреки че физически са свързани към една и съща мрежа.

- Мрежи за споделяне на ресурси

Такива обикновено са офисните компютърни мрежи, в които група хора споделят общи ресурси – файлове, принтери, документи, независимо дали тези

ресурси се намират на сървър, на локалните компютри или на специализирано мрежово устройство. Възможно е да съществуват правила с различни нива на достъп, които да определят кой потребител с кой ресурс може да работи. Такива мрежи често имат и достъп до Интернет, но ресурсите в мрежата трябва да са достъпни само за служителите вътре в мрежата и се вземат допълнителни мерки за защитата им от външни потребители. Често Интернет достъпът в такива мрежи може да бъде ограничен до определени места или услуги.

- Корпоративни мрежи

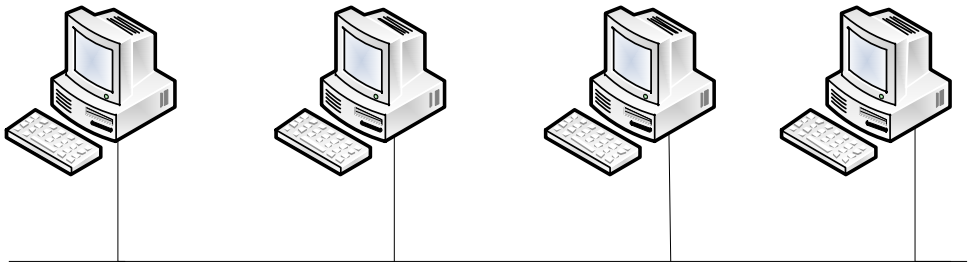
Корпоративната мрежа свързва отделните офиси и устройства на една организация в една обща мрежа, така че всички служители да могат да обменят информация помежду си, независимо в кой офис и отдел се намират. В корпоративната мрежа на защитата от външен достъп до документите на компанията се отделя особено голямо внимание. При свързване с отдалечени офиси се използват технологии за глобални мрежи, а често информацията се предава през Интернет, но за да се защити от външните потребители обикновено се използват различни механизми за сигурност, например виртуални частни мрежи (Virtual Private Networks, VPN).

### 1.3.4 Категоризация според топологията.

Топологията на мрежата представлява начина, по който са свързани компютрите в нея. Разглеждайки видовете топологии и техните характеристики, трябва да се има предвид, че една мрежа има два вида топологии. Първата е физическата топология, която определя как физически са свързани компютрите с преносната среда. Втората е логическа топология, която определя как се чувстват компютрите спрямо останалите в мрежата. В някои мрежи физическата и логическата топология съвпадат, в други се различават. За по-лесно описание долните примери разглеждат физическа топология.

- Топология шина (bus) – при нея всички компютри споделят обща преносна среда. Когато един предава, неговият сигнал се приема от всички останали, свързани към шината. Затова само един може да предава в даден момент от време. При прекъсване на шината, цялата мрежа не работи. Тази топология е типична за старите Ethernet мрежи с коаксиален кабел, за кабелните телевизионни мрежи и за някои съвременни оптични мрежи, изградени без отказоустойчивост.

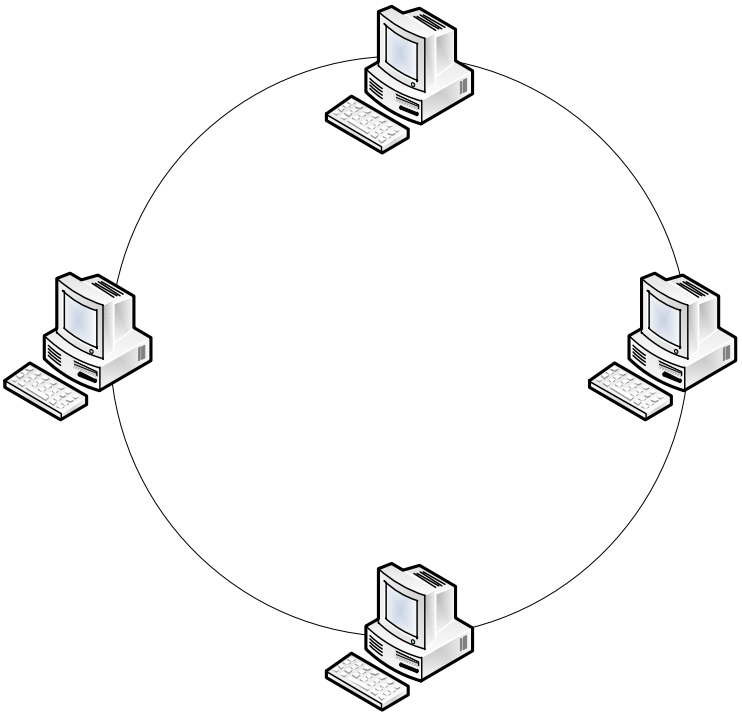
Пример за шинна топология е показан на фигура 1.5.



**Фигура 1.5. Шинна топология.**

- Топология кръг (Ring) – При нея кабелът от първия компютър се включва към втория, от втория в третия и така до последния, чиито кабел се включва обратно в първия. Когато един предава, неговият сигнал се приема само от следващия по кръга. Тази топология се отличава със отказоустойчивост, тъй като при прекъсване на кръга, все още има възможен път, по който да се предават данни между всички устройства. Такава топология имат мрежите от тип Token Ring, FDDI и съвременните оптични мрежи, изградени с отказоустойчивост.

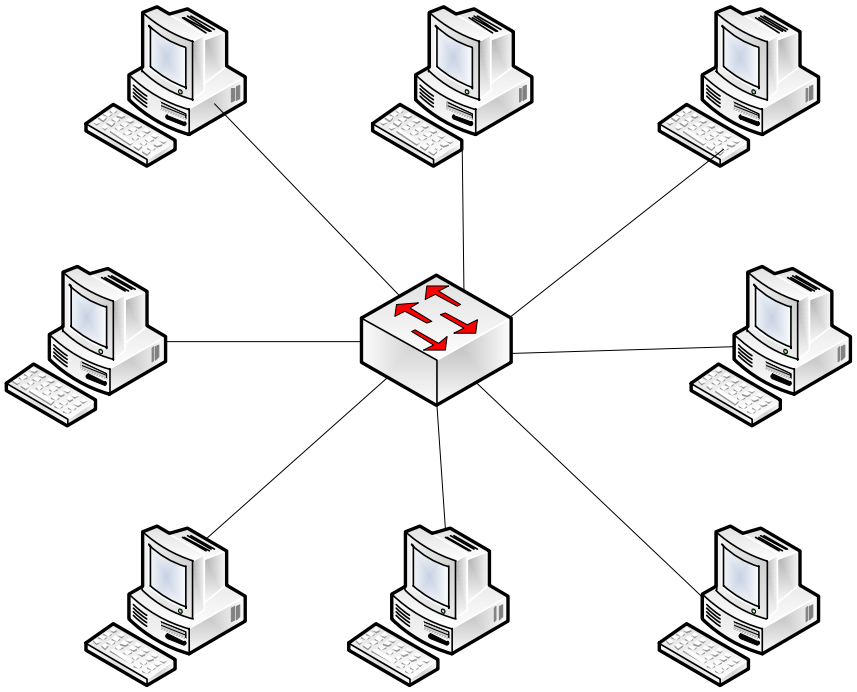
Пример за кръгова топология е показан на фигура 1.6.



**Фигура 1.6. Кръгова топология.**

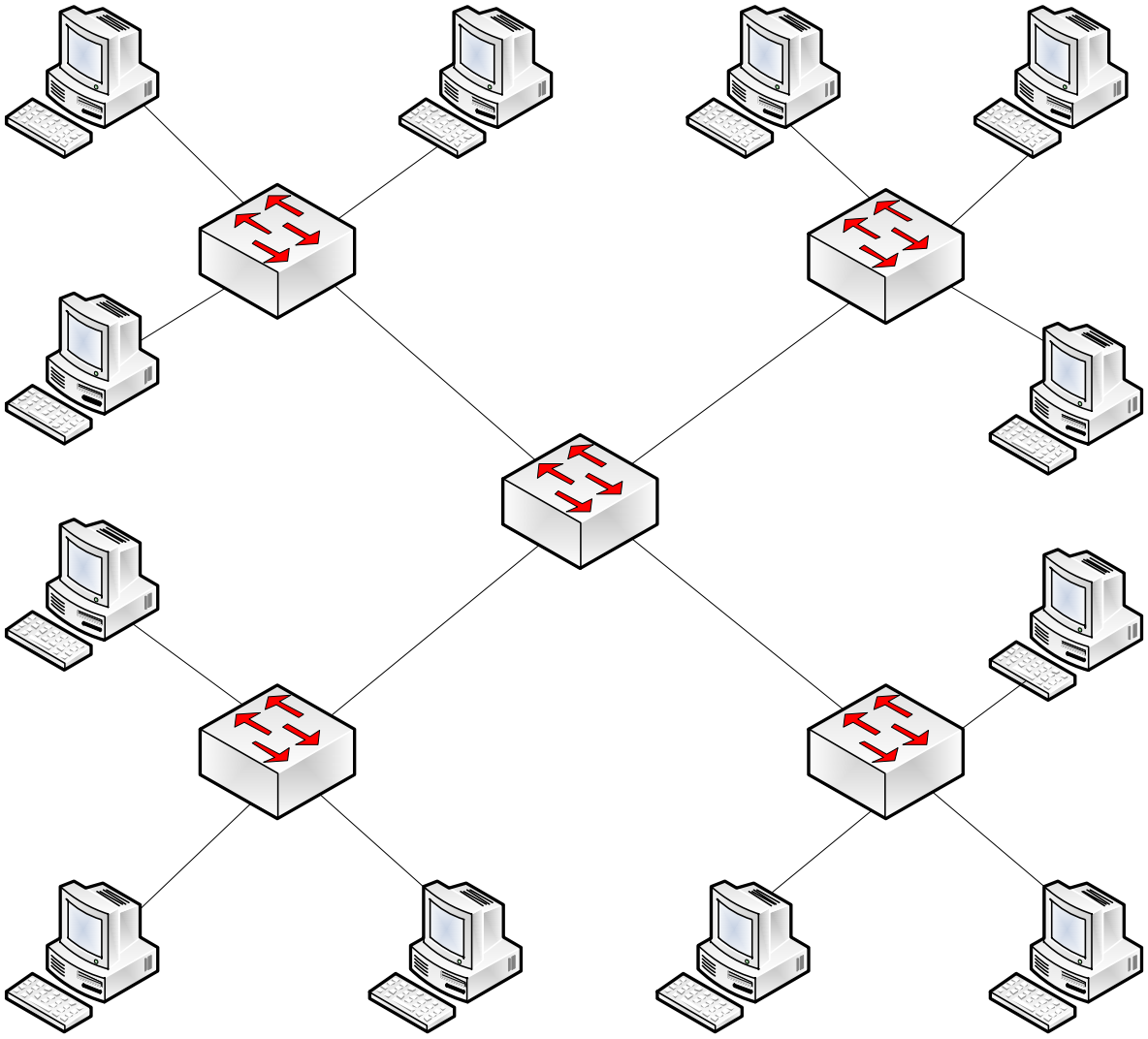
- Топология звезда (Star) – при нея има централен възел, към който са свързани всички компютърни системи, всяка чрез собствена среда. Така всеки компютър може да предава самостоятелно, без да се съобразява с останалите. Прекъсването на дадена среда влияе само на конкретния възел и не пречи на останалите. Такава мрежа е уязвима при отказ на централното устройство -

тогава цялата мрежа спира. Тази топология е типична за съвременните малки локални мрежи и за корпоративни мрежи, при които всеки от отдалечените офиси е свързан само към централата. Пример за звездообразна топология е показан на фигура 1.7.



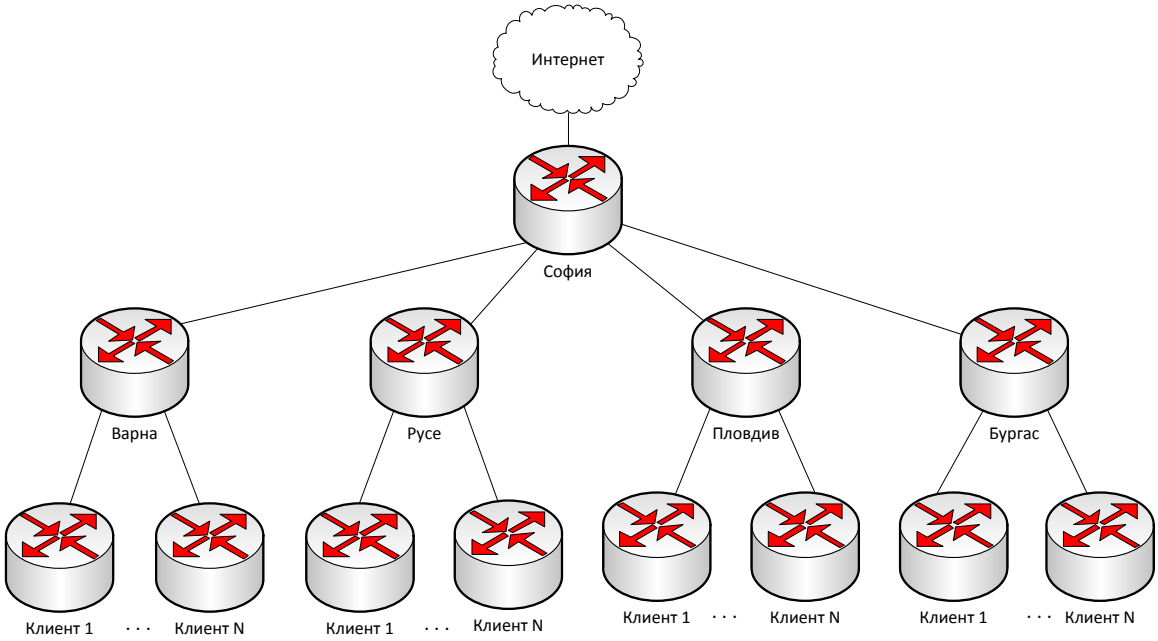
**Фигура 1.7. Звездообразна топология.**

- Топология разширена звезда (Extended star) – при нея някои от краищата на основната зведа стават центрове на свои звездообразни топологии. Структурата е типична за големите локални мрежи, например в многоетажна сграда една звезда свързва отделните етажи, където се поставя ново устройство (комутатор), разпределящо връзките до отделните стаи на етажа. Пример за топология разширена звезда е показан на фигура 1.8.



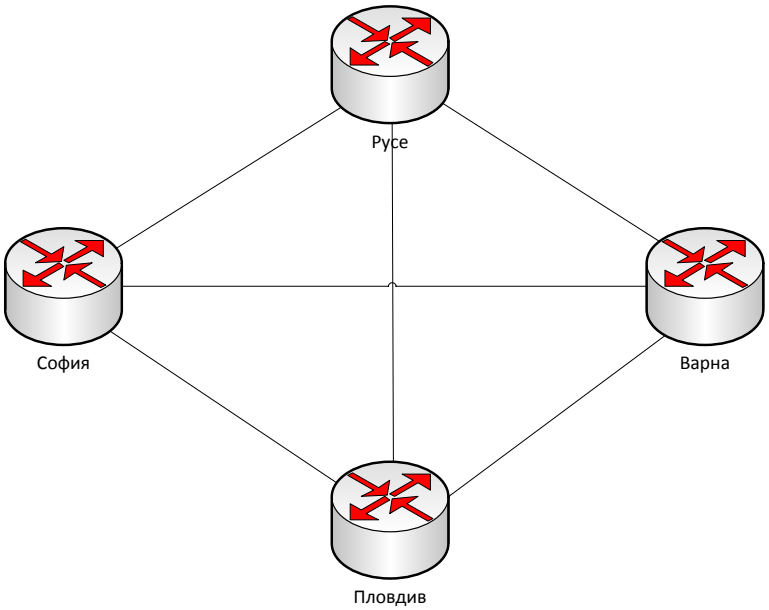
**Фигура 1.8. Топология разширена звезда.**

- Йерархична топология (Tree) – йерархичната или дървовидна топология има главен възел, към който се свързват неговите подчинени възли. Всеки от подчинените възли може да бъде главен за нови възли на по-долно ниво. Структурата е типична за мрежа на доставчик на Интернет услуги, при който от главния град, където е връзката му към Интернет има отделни връзки за останалите градове, където ще има клиенти, а от съответния град се изграждат нови връзки към клиентите. Топологията се характеризира с липса на отказоустойчивост – когато един възел или връзка отпадне, всички възли под нея губят достъп до мрежата. Пример за йерархична мрежа е показан на фигура 1.9.



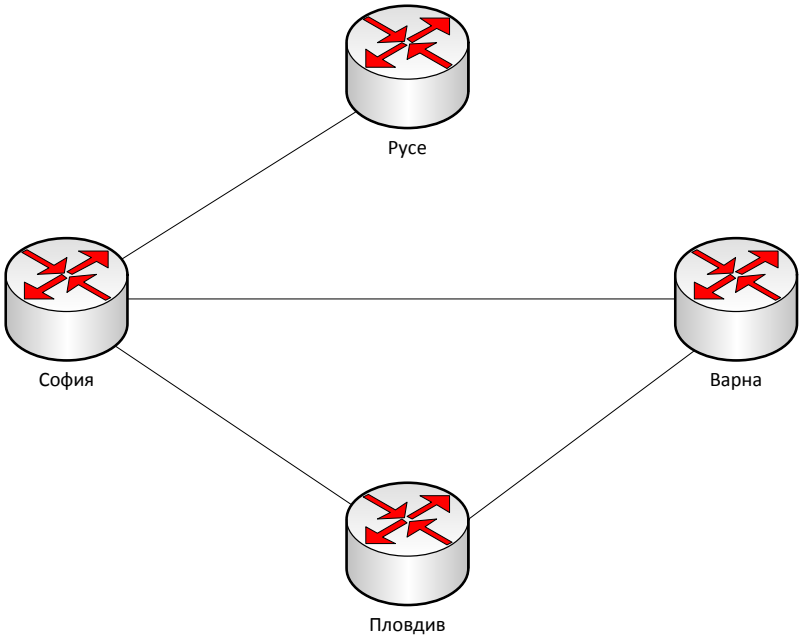

**Фигура 1.9. Йерархична топология.**

- Напълно свързана топология (Mesh) – при нея всеки възел има независима връзка със всеки друг. Предимството на топологията е голямата степен на отказоустойчивост, защото има голям брой резервни трасета, по които може да се предава информация, при отпадане на основното трасе. Недостатък на топологията е големият брой необходими връзки, което води до значително повишаване на цената, особено при голям брой възли. Структурата е типична за компания с клонове в различни градове, която иска да има отлична отказоустойчивост. Пример за такава топология е показан на фигура 1.10.




**Фигура 1.10. Напълно свързана топология.**

- Непълно свързана топология – тя няма точно определена структура, но целта е да се използва предимството на отказоустойчивостта на напълно свързаната топология, но да се намали цената. Затова важните възли се дублират с резервни трасета, за да бъдат отказоустойчиви, а по-маловажните за работата на компанията се оставят с по-малък брой връзки, за да се намали цената. Всяка неправилна топология може да се превърне в напълно свързана, при добавяне на необходимите връзки. Възможен пример за непълно свързана топология е показан на фигура 1.11.




**Фигура 1.11. Непълно свързана топология.**

# 2. Модели на компютърните мрежи

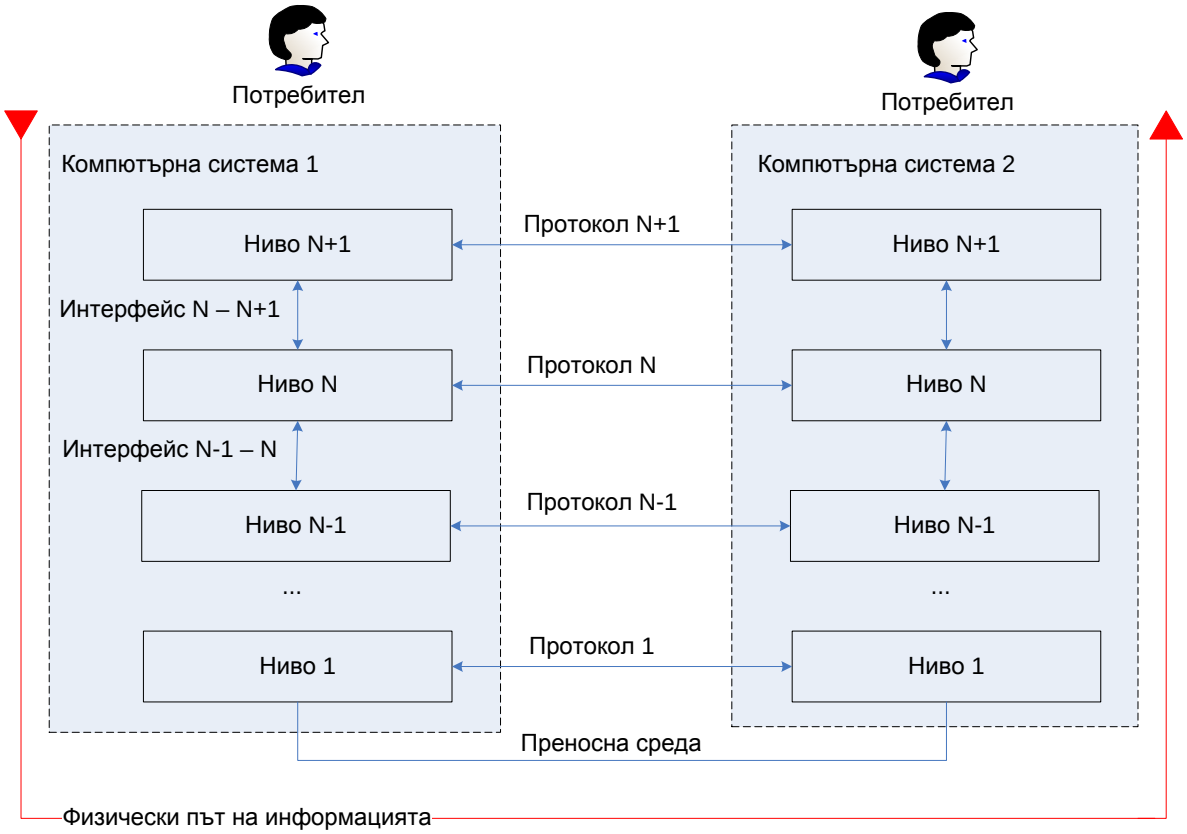
## 2.1 Разделяне на моделите на нива.

В началните години на компютърните мрежи не са съществували стандарти, които да определят как точно трябва да се построи и как да работи една компютърна мрежа. Тогава повечето големи производители са разработвали свои собствени решения за свързване на компютърните си системи в обща мрежа. Това разбира се е водило до трудности и несъвместимости при свързване на системи (софтуерни и хардуерни) от различни производители.

Тези трудности са обосновали нуждата от разработка на стандарти и модели в областта на мрежовите технологии. Тъй като компютърната мрежа е сложен комплекс, състоящ се от преносна среда, хардуер, операционна система, системен и приложен софтуер, то се налага проблемите за стандартизация на комуникацията да се разделят в отделните области, съставящи системата. Затова повечето модели на функциониране на компютърната мрежа представят комуникацията разделена на отделни нива, като всяко от нивата има определени функции.

Разделянето на модела на нива помага да се съсредоточат усилията на специалистите върху проблемите от своята област, без да се налага да се обръща особено голямо внимание на останалите компоненти от системата. По този начин се осигурява стандартизация на комуникацията, възможност за универсална комуникация, независимо от хардуерна платформа, операционна система или приложен софтуер. Типичен пример за комуникация, основана на общи стандарти е съвременната комуникация в Интернет – независимо дали работим с компютърна система, таблет, телефон или дори с телевизор, независимо дали операционната ни система е Microsoft Windows, някоя дистрибуция на Linux, Google Android или Apple Mac OS, дали използваме Internet Explorer, Mozilla Firefox, Chrome или Opera, ние имаме възможност да отваряме Web страници в Интернет, да четем и изпращаме електронна поща, да обменяме файлове и снимки, да играем игри в мрежата и всички останали функции, които използваме. Всички тези действия могат да бъдат извършвани по идентичен начин, независимо от начина ни на връзка – дали е през кабел за свързване към компютърна мрежа, по безжичен (Wi-Fi) начин или през Bluetooth.

На фигура 2.1 е показан обобщен модел, разделен на нива.




**Фиг. 2.1. Комуникация на нива.**

Физическият път на информацията е между двамата потребители, т.е. ако си представим, че единият изпраща електронна поща на другия, то физически информацията тръгва от най-горното ниво на първата система (програмата за електронна поща на първия потребител), преминава през всичките нива на тази система, преминава през преносната среда на мрежата, достига до втората компютърна система, там преминава през всички нейни нива отдолу нагоре и накрая достига до другия потребител (например под формата на електронна поща). Но понеже всяко от нивата изпълнява определени функции, то трябва да изпраща някаква информация на отсрещното ниво. Примери за такава служебна информация могат да бъдат: потребителско име и парола за системата за електронна поща, адреси на компютърните системи, потвърждения за правилно получена информация и други. По този начин обикновено между всеки две еднакви нива на системите се осъществява логическа комуникация, която е означена на фигурата с хоризонтални стрелки.

Стандартът, които определя обмена на информация между две съседни нива в компютърната система се нарича интерфейс. Обикновено между всеки две съседни нива има стандартизиран един интерфейс. Благодарение на интерфейсите можем да сменим част от нашата система – например средата за комуникация и мрежовия контролер (от кабел на безжичен достъп) и да продължим да използваме същата услуга от мрежата, без да сменяме останалите нива – например операционна система и програма.

Стандартите за обмен на данни между едноименните нива от всяка система се наричат протокол. Видно от фигурата е, че на всяко от нивата обикновено работи различен протокол. В доста примери за протокол се дава аналогията с езиците, които говори един човек – за да се разберат двама души, те трябва да владеят в някаква степен поне един общ език. По същия начин за да обменят информация две компютърни системи, на всяко едно от нивата те трябва да имат инсталиран поне един еднакъв протокол. Аналогията с езиците обаче може да ни подведе, защото при хората се счита, че колкото повече езици владее един човек е толкова по-добре, защото с по-голяма вероятност той може да се разбере с отсрещния. При компютърните системи обикновено се счита за лош подход инсталирането на всички възможни протоколи на всяко от нивата. Въпреки че по този начин те стават по-съвместими с останалите, този подход води до повишено натоварване на компютърната система и на мрежата, а понякога и до понижена сигурност.

Съществуват два подхода за осъществяване на комуникация, когато едната система няма знания на дадено ниво за протокола (протоколите) на другата система:

- На едната от системите да се инсталира протокола, познат на другата и по този начин първата да получи знания за правилата за комуникация с отсрещната страна;
- Между двете системи да се постави междинна система, която да познава и двата протокола и да „превежда“ от единия протокол на другия и обратно.

В повечето съвременни реализации се предпочита първият подход. Вторият е бил широко използван в миналото, когато повечето операционни системи са имали свои набори протоколи и за осъществяване на еднотипна комуникация между две различни системи (например електронна поща) се е налагало въвеждането на междинна система. Този подход обаче ще намира все по-голямо приложение в бъдеще време, защото в Интернет предстои една голяма промяна – преходът от сегашния набор протоколи – IP версия 4 към новия набор – IP версия 6. Повече детайли за нуждата, ползите и евентуалните предизвикателства при този преход са показани в главата за протокола IP версия 6 (IPv6), където е обяснен и детайлно единия възможен механизъм за преход, при който се предполага междинна система да превежда от стария към новия протокол, с което да направи възможна комуникацията на старите системи към новия Интернет.

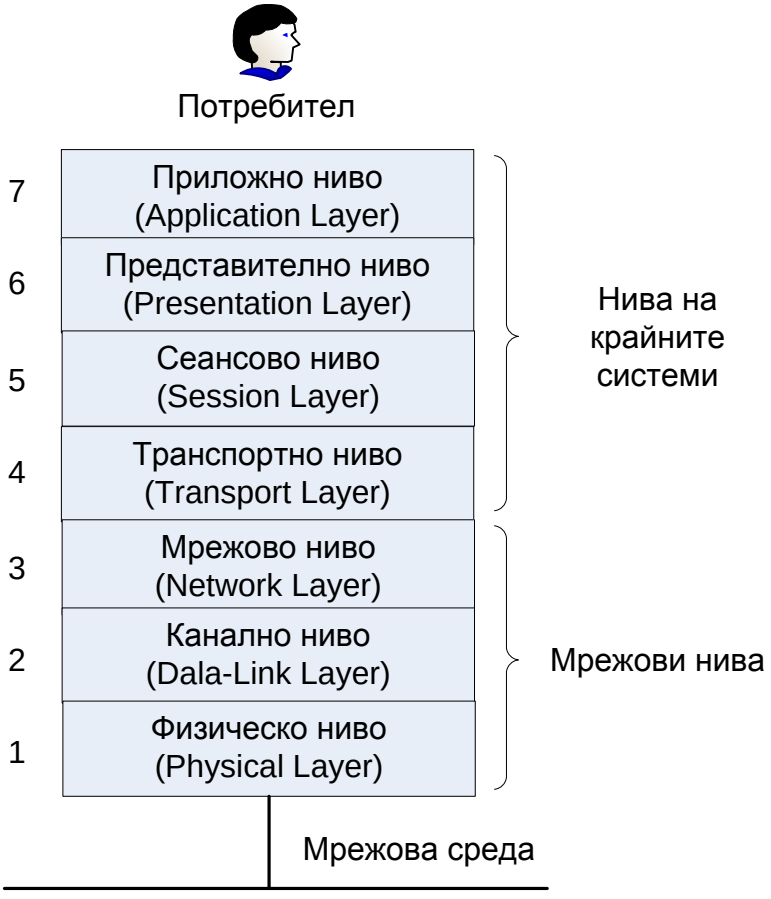
## 2.2 Препоръчителен модел на ISO – модел на взаимодействие между отворени системи – OSI (Open System Interconnection)

Един от основните модели, използвани в съвременния мрежов свят е съставен от международната стандартизационна организация ISO през 1984 година и се нарича „Препоръчителен модел за взаимодействие между отворени системи“ (Open System Interconnection reference model). Понятието „отворена система“ означава система,

изградена според препоръките и е отворена за комуникация с други системи, също изградени според същите препоръки.

Въпреки че повечето съвременни системи имат за цел да бъдат „отворени“, за да могат да си комуникират с останалите, практиката познава изграждането и на „затворени“ системи, изпълнявайки мрежова комуникация чрез нестандартна, скрита за света съвкупност от интерфейси, протоколи и правила за комуникация. Разбира се, целите на подобни системи са повишаване на степента на сигурност, запазване на авторски права или осигуряване на възможност за комуникация в такава мрежа само на системи, разработка на дадена фирма. Това противоречи на принципите на съвременния мрежов свят, чиято цел е да осигури възможност за еднаква комуникация навсякъде и със всякакви средства, но може да помогне на съответните компании за получаване на по-голяма печалба от разработените от тях системи.

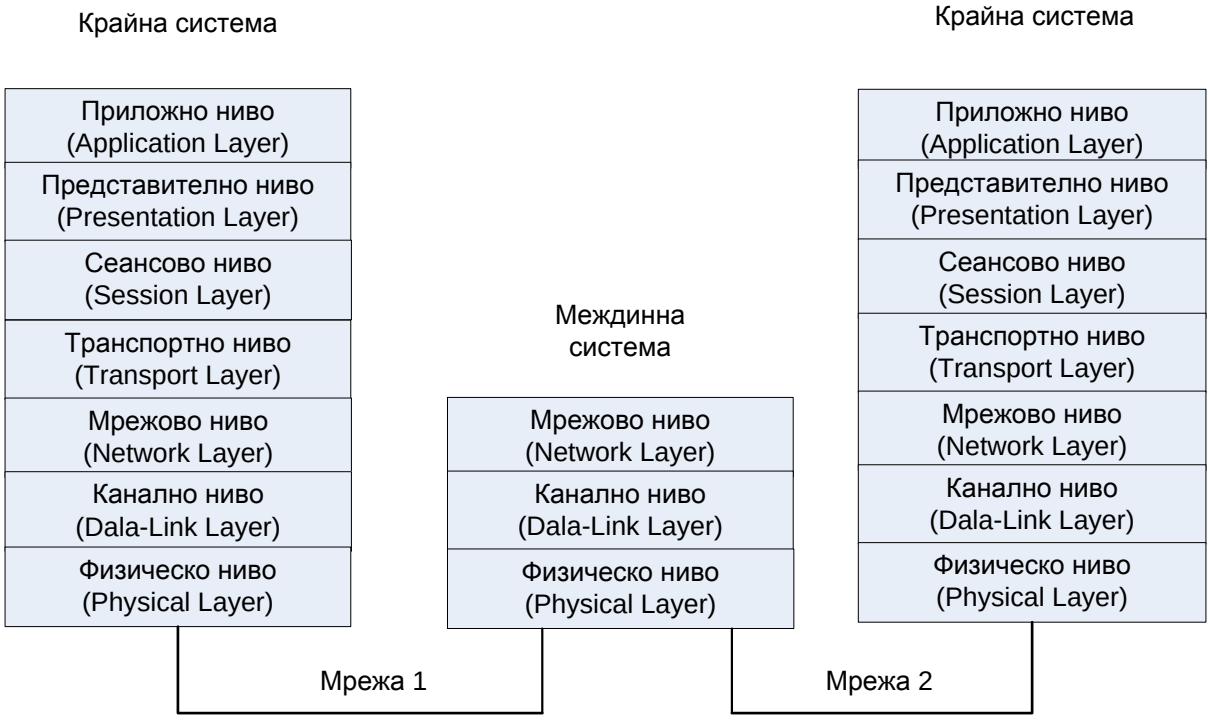
Моделът на ISO (често наричан OSI модел) е структура на седем нива. Техните наименования и местоположение са показани на фигура 2.2.[2]




**Фиг. 2.2. Препоръчителен модел на ISO (OSI модел)**

Въпреки че физически комуникацията започва от най-горното ниво на източника и преминава през всички, достигайки до най-долното, нивата обикновено се номерират от долу нагоре, т.е. най-долното ниво е с номер 1, а най-горното – с номер 7. В

настоящата книга нивата ще бъдат наричани с техните имена, вместо с номерата, но в много други публикации се използва номерацията на нивата. Въпреки че препоръката е изграждането на компютърната система със седем нива, не всички системи изпълняват всички възможни функции и затова не на всички са необходими всичките нива. На фигура 2.3 е показан традиционен модел на комуникация в Интернет.




**Фиг. 2.3. Традиционна комуникация в Интернет**

При тази комуникация мрежите са свързани помежду си чрез междинни системи – маршрутизатори (routers). Тези системи не изпълняват всички функции, а имат само необходимия набор, чрез който да намерят къде е получателят на информацията и да осигурят предаването до него. Затова традиционно тези системи са изградени до трето (мрежово) ниво, а останалите нива са налични само в крайните системи свързани към мрежата и използващи нейните услуги. По тази причина често долните три нива се наричат „мрежови нива“, защото те осигуряват услугите на мрежата и самата мрежа е изградена до тях, а горните четири нива се наричат „нива на крайните системи“, защото се срещат само в тези системи, използващи мрежата за своите цели на комуникация. Следва кратко описание на отделните нива, които ще бъдат по-подробно разгледани в следващите глави.

### 2.2.1 Физическо ниво (Physical Layer)

Това ниво се грижи за предаването на информацията по мрежовата среда. То формира битовете 1 и 0, използвайки различни физически свойства на средата – електрически импулси по металните кабели, светлинни импулси по оптичните влакна

или радиовълни при безжични предавания. Физическото ниво се изпълнява хардуерно и представлява съвкупността от преносна среда, съединителни елементи (конектори) и аналоговата част на мрежовия контролер, която физически предава и приема сигналите.

### 2.2.2 Канално ниво (Data-Link Layer)

Каналното ниво се грижи за доста богат набор функции. То определя дисциплината на достъп до средата – кой кога има право да предава данни. На канално ниво се прави и разграничаване на отделните компютърни системи, свързани към обща среда, осъществявайки адресиране чрез така наречените MAC адреси. Това помага в повечето случаи едно предаване на данни да се обработва само от системата, която трябва да получи тези данни, а не от всички компютърни системи в мрежата. Тук се определя и логическата топология на мрежата, която в някои случаи съвпада с физическата, а в други се различава. Каналното ниво може да осъществява някои функции за обработка на грешки при предаването на сигналите по физическата среда, за регулиране на скоростта на предаване на данните и други допълнителни функции, описани по-подробно в четвърта глава.

Каналното ниво е границата между хардуера и софтуера в компютърната система. Неговата долна част се реализира от цифровата част на мрежовия контролер и тя осигурява логиката за приемане и предаване на данни от и към физическото ниво. Горната част на каналното ниво обикновено се реализира софтуерно в драйвера на мрежовия контролер и така се осъществява интерфейса с операционната система.

### 2.2.3 Мрежово ниво (Network Layer)

Мрежовото ниво има различни функции в крайните системи, които предават данни чрез мрежата и в самите мрежови устройства, които трябва да осигурят възможността за предаване на тези данни между отделните системи.

Крайните системи използват функцията на мрежовото ниво – адресиране, за да се разграничават уникално в целия набор свързани мрежи. Например в Интернет всяка система трябва да има уникален адрес, наречен IP адрес. Чрез него различните системи разбират до кого трябва да предадат дадена информация, за да използват или да осигурят дадена услуга.

Междинните системи в Интернет използват IP адресите, за да определят най-добия път, по който трябва да премине изпратената от източника информация, за да достигне до получателя. Тази функция на мрежата се нарича маршрутизация (routing), а устройствата, които я осигуряват – маршрутизатори (routers). За да могат те да изпълняват тази функция, обикновено те си предават служебна информация за състоянието на различните мрежи и възможните пътища до тях, посредством маршрутизиращи протоколи.

На мрежово ниво могат да се изпълняват и други функции, например изпращане на съобщения за грешки при предаване и обработка на информацията в междинните

възли, както и изпращане на диагностични съобщения за проверка и откриване на неизправности в мрежата.

Мрежовото ниво в традиционните системи се изпълнява изцяло софтуерно, като в повечето реализации то е част от самата операционна система. Функционалността му може да се осигурява и чрез инсталиране на мрежови протоколи към дадена операционна система. Мрежовото ниво е ключово за разбиране и диагностика от страна на мрежовия специалист, тъй като обикновено компютърната мрежа завършва с това ниво. Това често означава, че повечето задължения на мрежовия специалист (администратор) завършват с осигуряване на правилното функциониране на мрежата до мрежово ниво, а останалите нива, които са част от крайните системи са задължение на системния администратор, който се грижи за операционните системи и приложните програми. Разбира се в някои организации е възможно един специалист да съвместява и двете функции.

### 2.2.4 Транспортно ниво (Transport Layer)

Транспортното ниво е нивото, което трябва да осигури надеждност на предаването на данни. Това е нивото, което разделя големите данни (файлове, електронни пощи) на порции, подходящи за предаване през мрежата. При разделянето нивото номерира отделните порции и така ги предава на мрежовото ниво, което се грижи да намери пътя за всяка от порциите му до получателя. Когато приемникът получи дадена порция информация, неговото транспортно ниво проверява номерацията на всички получени порции, подрежда ги в правилния ред, чрез изпращане на потвърждения се грижи за повторното изпращане на неполучената и сгрешена информация и накрая сглобява получените порции, предавайки към горните нива получените цели данни под формата на файлове, електронна поща или web страница.

Транспортното ниво осигурява и многозадачността на компютърните системи при работата им с мрежата. Не е необичайно една компютърна система работейки в мрежа едновременно с различни програми да отваря web страница, да получава файл, да чете електронна поща и да изпълнява други мрежови функции. Това означава, че в даден интервал от време до нашата компютърна система ще пристигат последователно порции от web страницата, от файла, от електронната поща и от другите приложения. Разпределянето на отделните порции информация към различните приложения се прави от транспортното ниво, посредством номерата на портове. Всяка програма, работеща на даден компютър получава уникален номер на порт. Чрез IP адреса на мрежово ниво, компютърът се идентифицира уникално сред останалите компютри в мрежата, а чрез номера на порта се идентифицира програмата на този компютър, която предава или приема данните. Комбинацията от IP адрес и номер на порт често се нарича сокет (socket).

Транспортното ниво най-често се изпълнява софтуерно, като част от операционната система, възможно е да се инсталира и като отделен протокол, заедно със съответния протокол от мрежово ниво. На транспортно ниво могат да се настройват доста

параметри, част от които могат да влияят на функционалността или производителността на компютърната система при работата и в мрежа.

### 2.2.5 Сеансово ниво (Session Layer)

Сеансовото ниво осигурява връзка (сеанс) между двете програми, които си предават информация. Например ако проверяваме електронната си поща, предварително нашата програма – клиент (например Microsoft Outlook, Mozilla Thunderbird, Apple Mail или дори web браузъра при web-базирана електронна поща) осъществява връзка със сървъра ни за поща. При установяването могат да се проверяват потребителско име и парола, права за достъп и други параметри. Сесията се поддържа активна, докато едната страна (клиентът или сървърът) я прекратят.

В повечето съвременни реализации сеансовото ниво, както и другите нива нагоре се реализират софтуерно и са част от програмите, които осъществяват дадена мрежова функционалност.

### 2.2.6 Представително ниво (Presentation Layer)

Представителното ниво има три главни функции. Първата, която дава и името му е представянето на данните по начин, който да бъде разбираем за двете крайни системи, които си предават информация. Например ако отваряме една web страница написана на кирилица, web сървърът може да сигнализира на браузъра за кодовата таблица, на която е написана страницата и тя автоматично да се покаже с кирилски букви, вместо с неразбираеми символи.

Другите две функции на нивото са компресирането на данните, така че те да се предават по-кратко време по средата за връзка и криптирането им, така че ако някой неоторизиран получи достъп до данните при тяхното предаване да не може да ги разчете за разумно дълъг срок.

### 2.2.7 Приложно ниво (Application Layer)

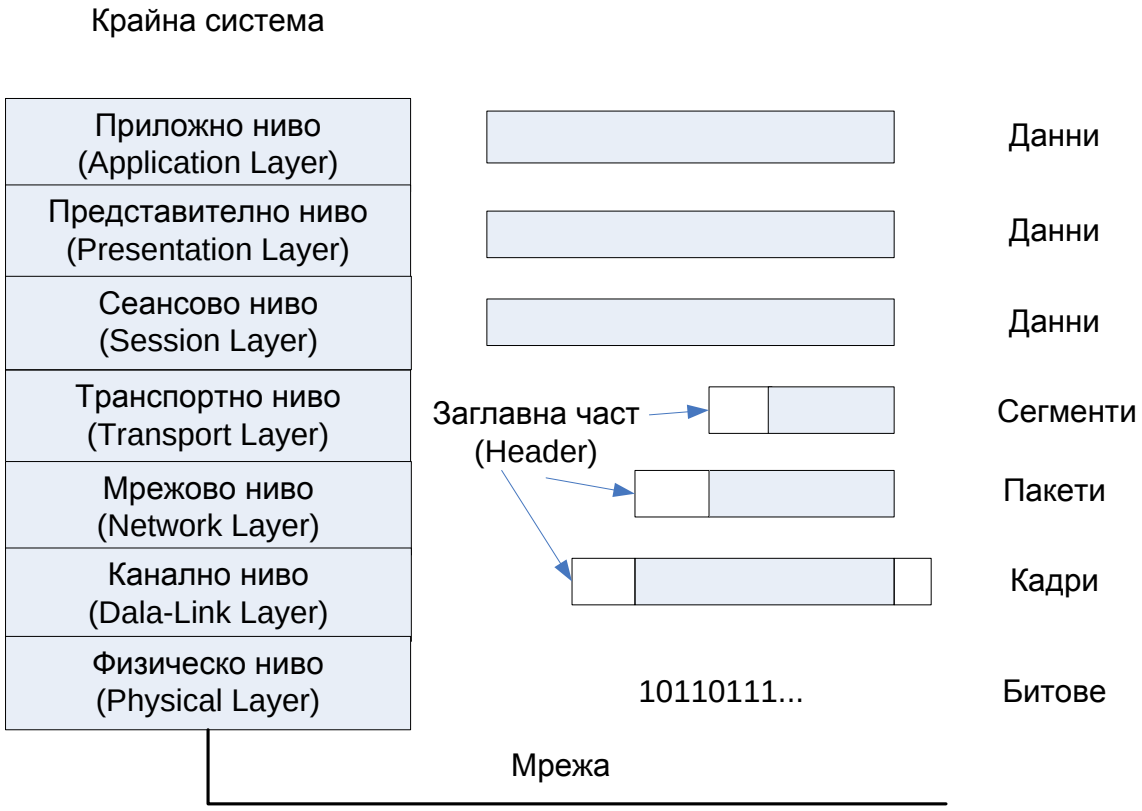
Приложното ниво осигурява мрежови услуги на потребителите и приложенията по стандартизиран начин, така че използвайки например услугите предаване и приемане на електронна поща да получаваме еднакво обслужване, независимо от клиента за електронна поща, който използваме или от типа на сървъра, към който се свързваме. Част от услугите, които осигурява нивото могат да бъдат прозрачни за приложенията, които нямат вградена мрежова функционалност – например да се осигури на един текстов редактор възможност да записва и чете файлове от мрежов диск.

## 2.3 Капсулация на данните (Data Encapsulation)

Данните преминават през различни трансформации при преминаването си през различните нива на модела. При повечето компютърни мрежи една голяма по обем информация (файл, електронна поща) не се предава наведнъж, а се разделя на

отделни порции. Това разделяне става на транспортно ниво. То взима поредната порция данни и добавя към нея своя заглавна част (header), която съдържа служебните за нивото данни, осигуряващи функционалността на нивото. В примера за транспортното ниво част от информацията, която се записва в заглавната част е например поредния номер на порцията и номера на порта, указващ програмата, за която е предназначен. Понеже тези служебни данни са предназначени само за отсрещното транспортно ниво, то долното мрежово ниво взима предадената от горното единица информация без да я интерпретира и към нея добавя своя заглавна част, която съдържа неговата служебна информация, например IP адресите.

Процесът, при който всяко ниво добавя своята служебна информация към порцията данни се нарича капсулация на данните (Data encapsulation) и е показан на фигура 2.4.



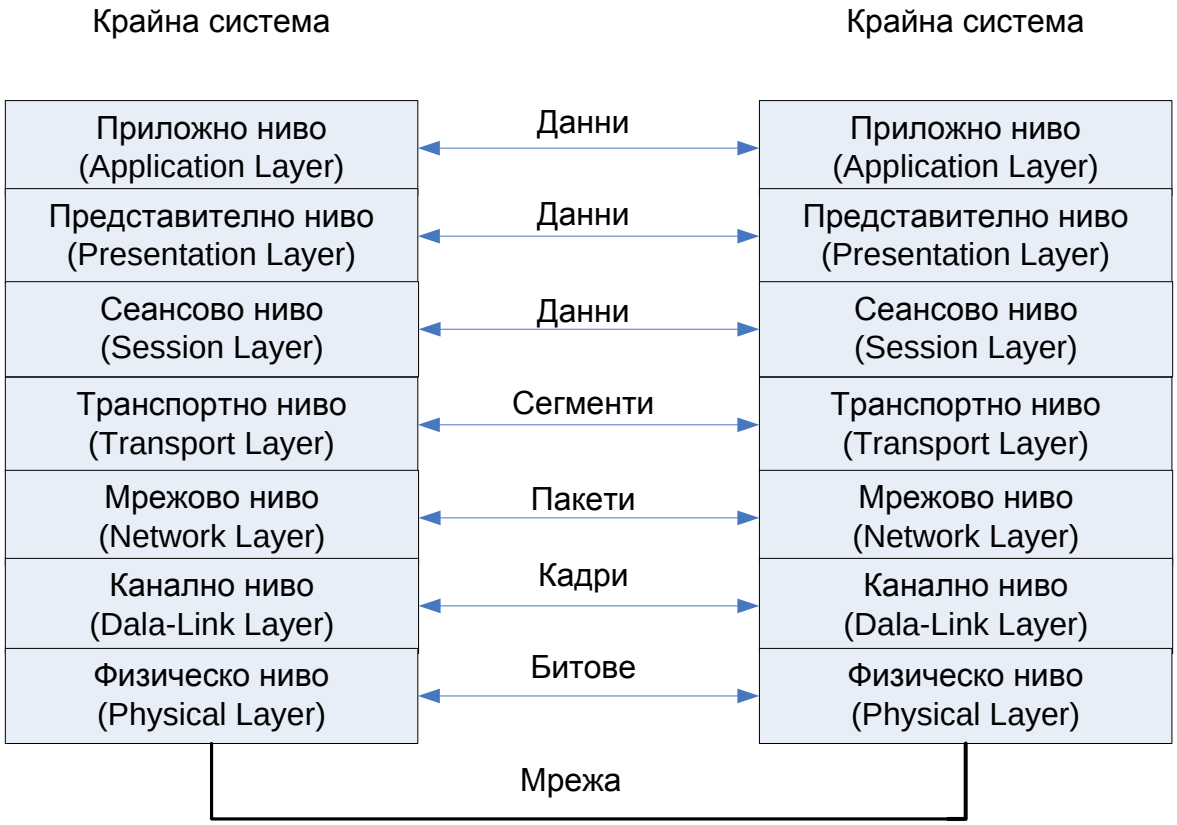

**Фиг. 2.4. Капсулация на данните**

Обикновено каналното ниво освен заглавна част добавя и служебна информация в края на предаваната порция данни, която служи за проверка за грешки и се нарича завършваща част (trailer).

Този процес показва, че порциите данни имат различен формат и съдържание, в зависимост от нивото, на което се намират. Когато говорим за горните три нива, то данните се наричат просто данни, защото за тях единицата информация, която се предава е един файл, една поща или една web страница, без тя да е разделена на порции. Транспортното ниво накъсва данните и добавя своята заглавна част към тях,

така тези порции се превръщат в сегменти (segments). Сегментът с добавена заглавна част на мрежово ниво се нарича пакет (packet), а на канално ниво единицата информация се нарича кадър (frame). Физическото ниво не добавя и не интерпретира информацията, която предава, а просто я превръща в поредица от битове (единици и нули) и така я предава по средата.

Наименованията на отделните порции данни в зависимост от нивото са показани на фигура 2.5.




**Фиг. 2.5. Единици данни на различните нива**

Когато именуваме порциите данни, без да уточняваме на кое ниво в момента се намират те, най-често използваме термина пакети, защото пакетът е единицата данни на мрежово ниво, а това е нивото на което работи компютърната мрежа.

### 2.4 Интернет модел

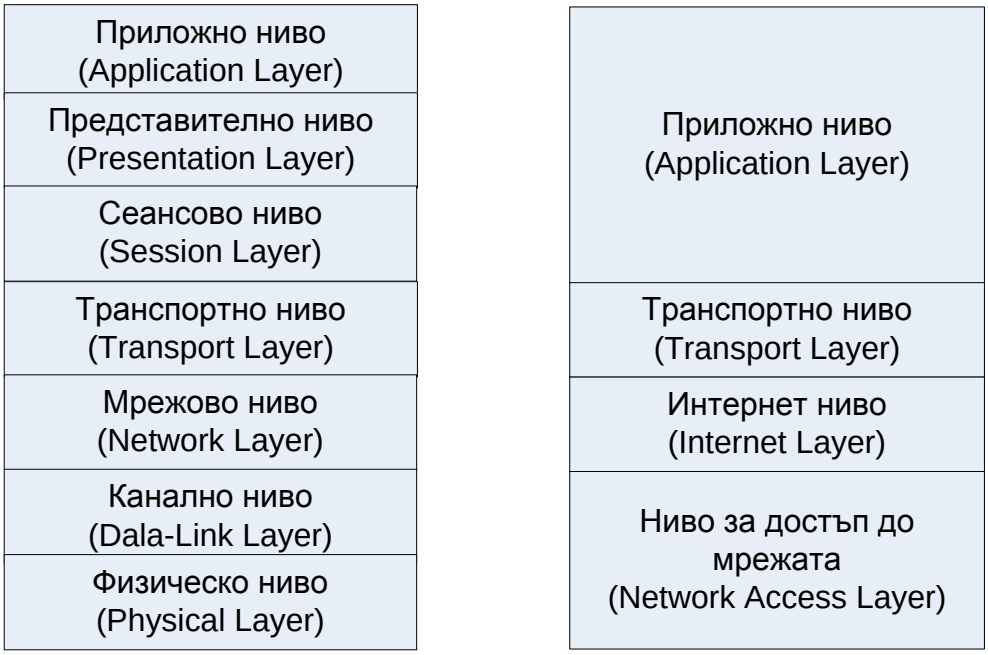
През 1969 година Министерството на отбраната на Съединените щати (Department of Defense, DoD) е разработило модел за комуникация, в началото реализиран в мрежата ARPANET, която в последствие е станала гръбнака на сегашния Интернет. Тази мрежа е спряна през 1990 г., но моделът с малки изменения е в основата на съвременните комуникации в Интернет. Този модел се нарича DoD модел, с името на разработилата го организация или Интернет модел с името на най-голямата мрежа, работеща по него или TCP/IP модел, с имената на главните протоколи, с които се

осъществява комуникацията. Моделът има четири, вместо препоръчваните от OSI седем нива и неговата структура е показана на фигура 2.6. [2]


**Фиг. 2.6. Интернет (TCP/IP) модел**

Тъй като по задание моделът е трябвало да може да работи във всякакви преносни среди и мрежови топологии, най-долното ниво се нарича „достъп до мрежата“ и там моделът не специфицира конкретни протоколи. Второто ниво – Интернет нивото определя функции, подобни на определените в мрежовото ниво на OSI и там работи главният протокол IP (Internet Protocol). Третото ниво се нарича транспортно или понякога ниво система-система (host-to-host) и там работят протоколите TCP (Transmission Control Protocol), който е разработен с цел да осигури надеждност на предаваните данни или UDP (User Datagram Protocol), който не осигурява надеждност, но е по-бърз и натоварва по-малко мрежата със служебни данни. Последното ниво на модела е приложното ниво, на което работят програмите (приложенията), използващи Интернет за комуникационна среда.

Сравнението на двата модела е показано на фигура 2.7.




**Фиг. 2.7. Сравнение на моделите.**

Най-долното ниво на Интернет модела съвпада с функциите, определени от физическото и каналното ниво на OSI. Мрежовото ниво на OSI съвпада по функции и местоположение с Интернет нивото, а транспортните нива са идентични, въпреки че според OSI транспортното ниво трябва да осигурява надеждност, а в Интернет има избор на протоколи дали да се осигурява надеждност или не.

Функциите на последните три нива на OSI са обединени в общото приложно ниво на Интернет, което означава, че самата мрежа не осигурява автоматично тези функции, а те са оставени за изпълнение на приложението, например ако е необходимо компресиране, криптиране на данни или проверка за потребителско име и парола, това се реализира не от мрежата, а от приложението.

Интернет моделът предшества OSI във времето, но той е важен, защото определя функционирането на най-голямата до момента компютърна мрежа – Интернет, която продължава да се развива с огромни темпове и практически достъпът до нея е навсякъде и функциите и се използват ежедневно от милиарди потребители по целия свят.

# 3. Физическо ниво

Физическото ниво е най-ниското ниво в модела. То се грижи за предаването на информацията по физическата среда на мрежата и за представянето на битовете по средата по такъв начин, че да може да се осигури необходимата скорост и шумоустойчивост на сигнала. Физическото ниво се състои от средата за предаване на информация, съединителите с които средата се свързва към контролера и частта от електрониката на платката, отговорна за представяне, предаване и приемане на сигналите.

## 3.1 Среди за предаване на информация и съединителни елементи.

### 3.1.1 Коаксиален кабел

Коаксиалният кабел е екранирана среда за връзка. Състои се от централно метално жило, пластмасова изолация, екранировка, изпълнена като оплетка от метални проводници и външна изолация (фиг. 3.1.).

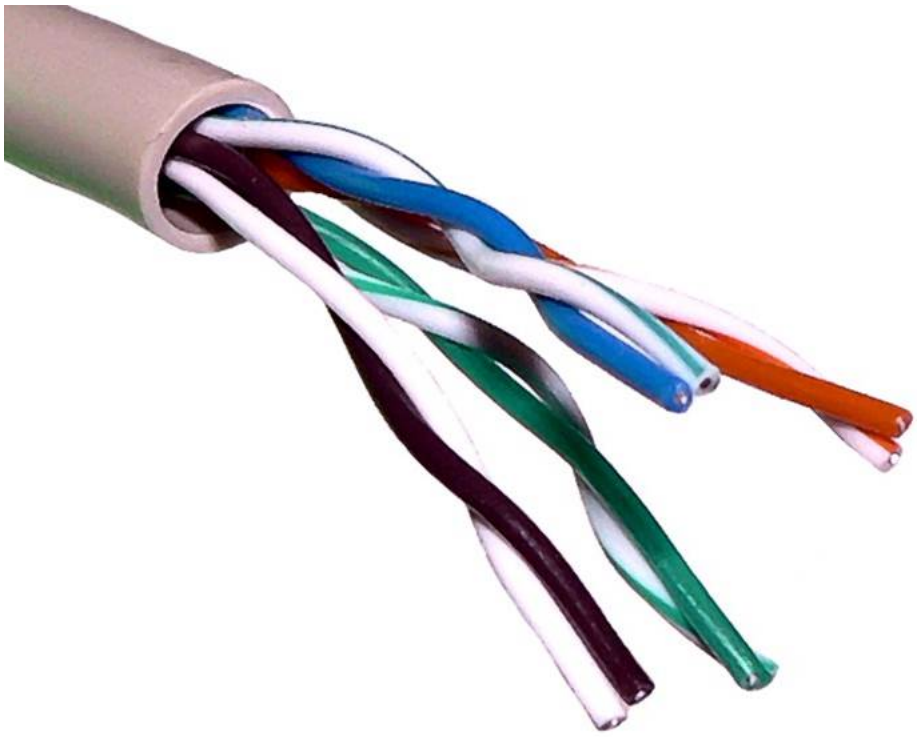
**Фиг. 3.1. Коаксиален кабел.**

В някои стари компютърни мрежи за преносна среда се използваха главно два типа коаксиален кабел: RG-59, наричан на жаргон „тънък“ (thin) и RG-8, наричан „дебел“ (thick). Тези мрежи имаха топология шина и работеха с протокол Ethernet –протоколът, използван и от съвременните мрежи с усукани двойки проводници и оптични влакна. Съединителите на тънкия кабел се наричат BNC (показани на фиг. 3.1), а на дебелия – N конектор.

В момента коаксиалните кабели се използват главно при свързването на антените на безжичните мрежи, при технологията за предаване на данни по кабелната телевизионна мрежа и за предаване на сигнал от аналогови видеокамери.

### 3.1.2 Усукани двойки проводници

Кабелите с усукани двойки проводници са най-често използваната среда при изграждане на съвременните локални компютърни мрежи. Кабелът усукана двойка проводници се състои от няколко (обикновено четири) двойки проводници, всяка от които е маркирана със определен цвят, като единият проводник от двойката е оцветен плътно с цвета, а другият е комбинация от бяло и съответния цвят. Обикновено цветовете са оранжев, зелен син и кафяв. На фиг. 3.2. е показан кабел неекранирана усукана двойка проводници (Unshielded Twisted Pair, UTP).

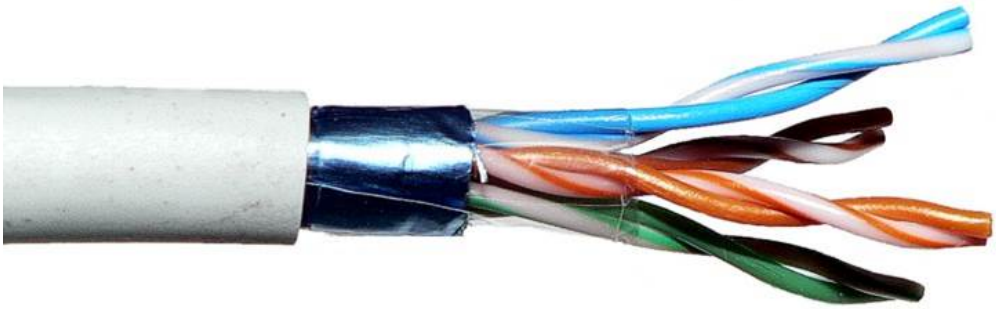


**Фиг. 3.2. Неекранирана усукана двойка.**

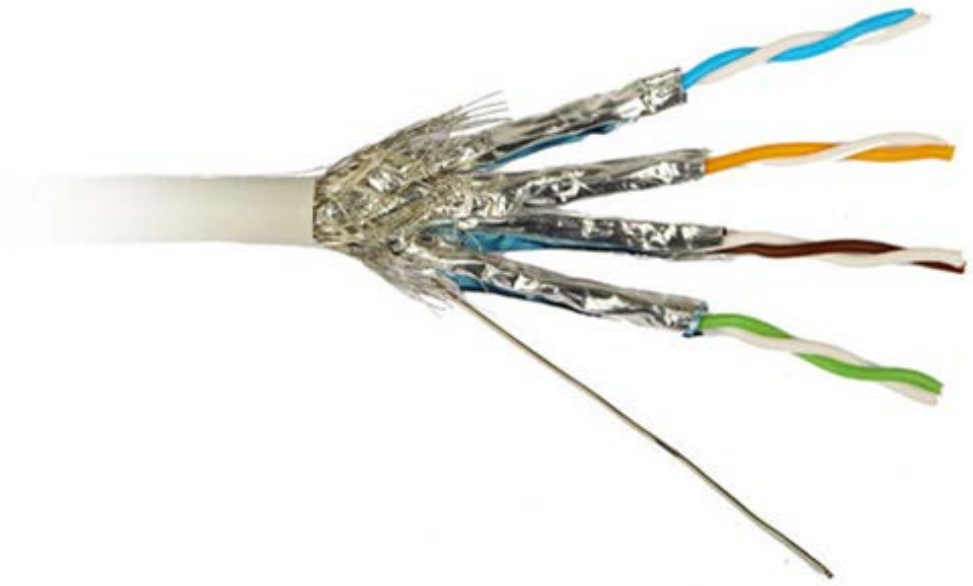
Двата проводника от всяка двойка са усукани помежду си. Усукването се прави, за да се защити сигнала, който се предава по двойката от влияние на външни смущения.

Поради това е неправилно при работа с този тип кабел да се предприемат всякакви механични действия, които могат да доведат до нарушаване на усукването – опъване, прегъване, настъпване на кабела и други, тъй като те могат да доведат до смущения при предаването на данните.

Въпреки усукването, при полагане в някои среди с големи електромагнитни излъчвания, например производствени помещения, в кабела могат да се индуктират напрежения, които да попречат на разпространението на сигнала. Затова съществуват екранирани варианти на кабела с усукани двойки проводници. Някои от тях имат единична екранировка – предпазвайки четирите двойки от външни смущения, а други имат двойна екранировка – всяка двойка е екранирана от останалите, а обща екранировка предпазва двойките от външни смущения. На фиг. 3.3 е показан единично екраниран кабел, а на фиг 3.4. – двойно екраниран.



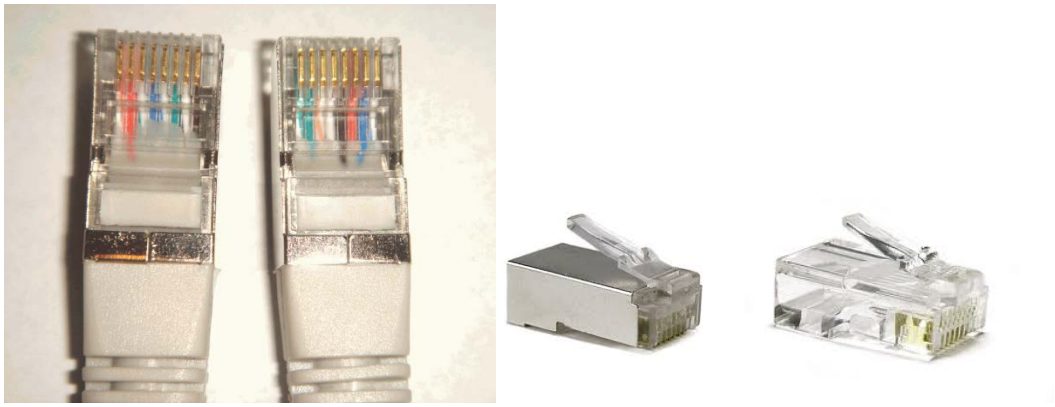
**Фиг. 3.3. Единично екранирана усукана двойка.**



**Фиг. 3.4. Двойно екранирана усукана двойка.**

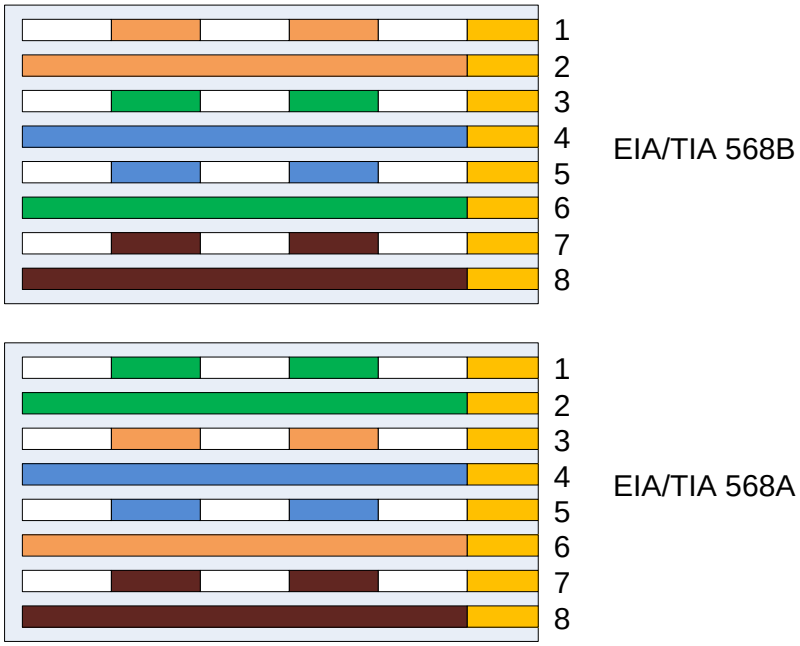
В някои случаи екранировката е изпълнена с метално фолио и тогава кабелът се нарича FTP (Foiled Twisted Pair), а в други – от оплетка от метални проводници и кабелът се нарича STP (Shielded Twisted Pair). Когато екранировката е двойна, първо се обозначава външната, а после вътрешната, например кабелът на фиг. 3.4 е SFTP (Shielded Foiled Twisted Pair).

Съединителите, използвани при усуканите двойки се наричат RJ-45 и имат осем контакта – за всеки от осемте проводника. При поглед от страната на проводниците първо перо е вляво, а осмо – вдясно. Съществуват екранирани и неекранирани – показани на фиг. 3.5.



**Фиг. 3.5. Съединители RJ-45 за усукани двойки проводници**

Подредбата на проводниците в съединителя е стандартизирана от стандарта EIA/TIA 568, като съществуват два варианта: EIA/TIA 568A и EIA/TIA 568B, показани на фиг. 3.6.



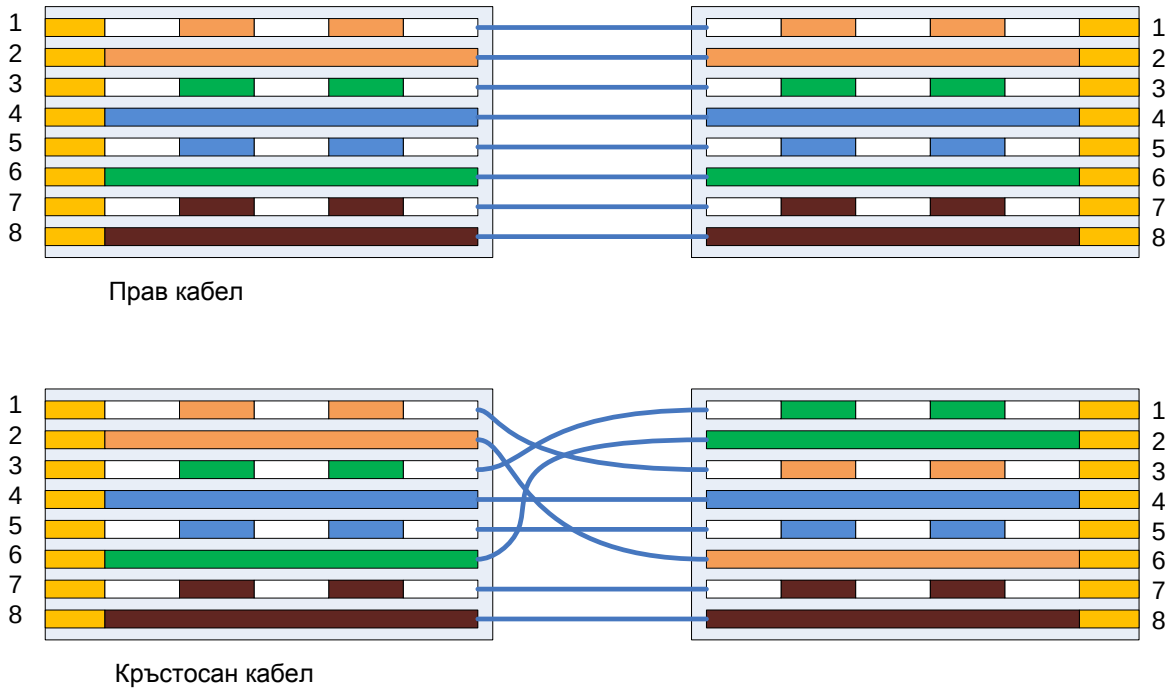

**Фиг. 3.6. Подредба на проводниците при усукани двойки.**

По-често в локалните мрежи се използва стандартът EIA/TIA 568B, при който проводниците се подреждат от първо към осмо перо по следния начин: бяло-оранжево, оранжево, бяло-зелено, синьо, бяло-синьо, зелено, бяло-кафяво, кафяво. При стандарта EIA/TIA 568A местата на оранжевите и зелените проводници са разменени.

Спазването на правилната подредба на проводниците е много важно, за да се осигури качествено предаване на данни.

Най-често изградените мрежи с усукани двойки се наричат Ethernet и имат звездообразна топология – към централно устройство, наречено комутатор (switch) се свързва всеки компютър със свой собствен кабел. При по-големи мрежи е възможно топологията да е разширена звезда – към един комутатор, например в една сграда да се свържат няколко други комутатора, намиращи се в отделни стаи, а към тях – отделните компютри.

Когато свързваме крайно устройство (компютър) към мрежово устройство (комутатор) се използва „прав“ кабел – кабел, при който накрайниците в двата края са поставени по един и същ стандарт, по-често EIA/TIA 568B. При свързване на еднотипни устройства, например два компютъра без комутатор помежду им или два комутатора понякога се налага да се използва т. нар. „кръстосан“ кабел – единият край е по единия стандарт, например EIA/TIA 568B, а другия край – по другия EIA/TIA 568A. Това разменя предавателите и приемниците на двете устройства и позволява директна комуникация. Двата типа кабели са представени схематично на фиг. 3.7.




**Фиг. 3.7. Прав и кръстосан кабел.**

Повечето съвременни устройства имат вградена функция Auto MDI/MDI-X, която се грижи за автоматичната размяна на приемна и предавателна двойка и тогава можем да работим само с прави кабели. Понякога тази функция не сработва коректно, особено при устройства от различни производители.

Кабелите с усукани двойки проводници имат характеристика – категория, която определя техните качества и възможности. В таблица 3.1 са показани категориите усукани двойки и най-често използваните стандарти за Ethernet мрежи.

**Табл. 3.1 Категории усукани двойки.**

| Категория | Технология | Скорост | Разстояние | Използвани двойки |
|---|---|---|---|---|
| 3 | 10BASE-T | 10 Mbit/s | 100 m | 2 |
| 5 | 100BASE-TX | 100 Mbit/s | 100 m | 2 |
| 5e | 1000BASE‑T | 1 Gbit/s | 100 m | 4 |
| 6 | 1000Base-TX | 1 Gbit/s | 100 m | 2 |
| 6a, 7 | 10GBASE‑T | 10 Gbit/s | 100 m | 4 |

Кабели категория 3 и 5 вече не се намират на пазара, но е важно да познаваме характеристиките им, защото все още могат да се срещнат инсталирани в някои сгради, където ако трябва да се повиши скоростта на компютърната мрежа може да се наложи да се смени и съществуващото окабеляване. В съвременните компютърни мрежи се използват кабели категория 5e за 1 Gbit/s или по-високи категории.

### 3.1.3 Оптични влакна

Оптичните влакна представляват светлопропусклива стъклена сърцевина (core), облечена в светлоотразяваща обвивка (cladding). По сърцевината се разпространяват светлинни, вместо електрически сигнали.

Оптичните влакна безспорно са най-перспективната среда за предаване на информация. Най-бързите в съвременния свят мрежови технологии – 40 Gbit/s и 100 GBit/s Ethernet, както и други технологии за глобални мрежи работят само по оптични влакна и за тях няма разработени алтернативи за метални проводници. Към момента на написване на тази книга има експериментални изследвания в лабораторни условия, които доближават или дори надвишават 100 Tbit/s (терабита, 1 терабит = 1024 гигабита) по едно оптично влакно, но има още време, докато те се стандартизират и излязат на пазара.

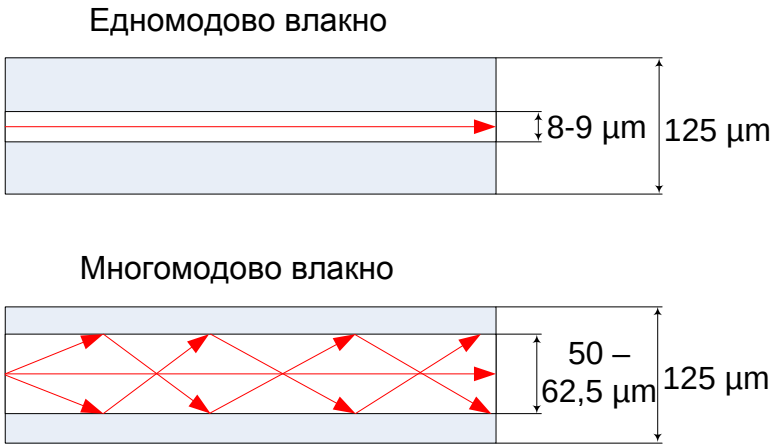
Оптичните влакна имат и други предимства – докато предаването по метални кабели достига до 100 метра за усукана двойка (по стандарт – в практиката е възможно постигане и на по-големи разстояния) и до няколкостотин метра при коаксиален кабел, при оптичните влакна е типично достигането на няколко десетки километра, дори има

технологии за над 100 километра. Тъй като в оптичното влакно няма метал, то не се влияе от външни смущения.

Недостатъкът на оптичните влакна е, че те са доста крехки и по-лесно могат да бъдат пречупени, така че за тях трябва да се положат по-специални грижи за предпазване от механични въздействия. На пазара съществуват специално заздравени оптични кабели, предназначени за подземно или въздушно полагане, които гарантират, че оптичните влакна в кабела няма да се пречупят при определени условия.

Тези характеристики определят най-честата употреба на оптичните кабели – за магистрални трасета на големи разстояния, свързващи няколко локални мрежи, а самите локални мрежи най-често се изпълняват с усукани двойки.

В момента най-често се използват два типа оптични влакна – показани на фиг. 3.8.

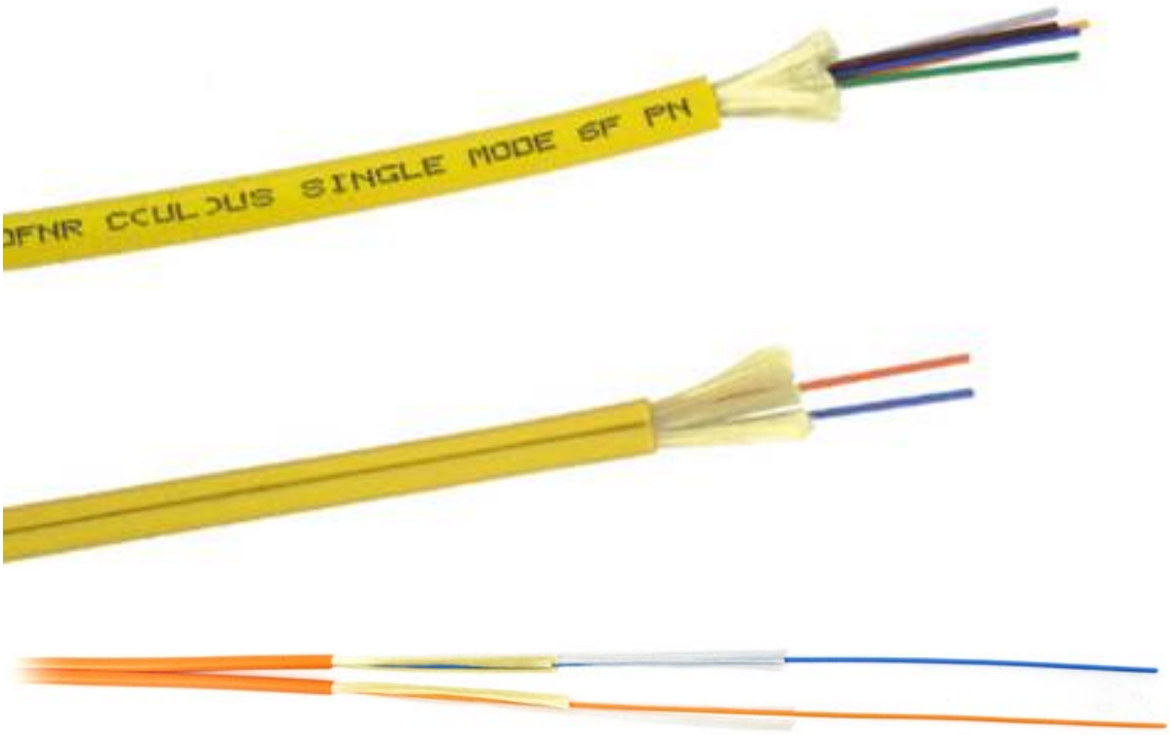



**Фиг. 3.8. Видове оптични влакна.**

Едномодовите (Single Mode) оптични влакна имат диаметър на стъклената сърцевина 8 или 9 микрометра. Външният диаметър на влакното и при двата вида обикновено е 125 микрометра. При едномодовите влакна светлината се излъчва от тясно фокусиран лазер, има една честота и се разпространява в права посока по дължината на влакното. Тези влакна постигат по-високи скорости на обмен и покриват по-големи разстояния, но светлинният източник е по-скъп, което повишава цената.

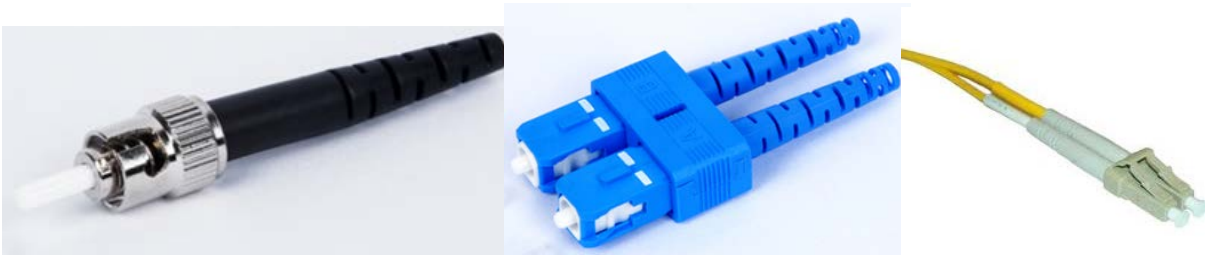
Многомодовите (Multi-Mode) влакна имат диаметър на сърцевината 50 или 62,5 микрометра. При тях източникът на светлина е специален лазерен светодиод, който излъчва сноп лъчи с близки дължини на вълните. Те се разпространяват отразявайки се от обвивката, при което губят част от енергията си. Поради това скоростите и разстоянията са по-ниски, но устройствата за предаване са по-евтини. Ако например по едномодово влакно може да се предава с 10 Gbit/s на разстояние над 100 километра, то с многомодово влакно могат да се постигнат около 550 метра.

Тъй като обикновено по оптичното влакно се предава сигнал еднопосочно (съществуват и технологии за двупосочно предаване), то обикновено в един оптичен кабел има няколко, до няколко десетки влакна. При някои е възможно с цел заздравяване влакното да е положено в пластмасова тръбичка, пълна с гел или да е обвито с текстилен материал – кевлар. Няколко вида оптични влакна са показани на фиг. 3.9.




**Фиг. 3.9. Оптични кабели.**

Съединителите, които се използват са няколко типа. По-старият вид са кръгли и се наричат ST, а по-новият са правоъгълни и се наричат SC. Напоследък в доста устройства се използват оптични модули с намален обем – SFP, който имат конектори от тип LC. Видовете съединители са показани на фиг. 3.10.



ST съединител SC съединител LC съединител

**Фиг. 3.10. Съединители за оптични влакна.**

Свързването на оптични влакна става чрез термично спояване (сплайс, splice). Цената на устройството е от няколко хиляди до десетки хиляди лева, затова ако не се използва постоянно, обикновено не е изгодно закупуването на устройство и организацията наема външен изпълнител за свързване на влакната. На фиг 3.11 е показана примерна машина за спояване на оптични влакна.

**Фиг. 3.11. Апарат за спояване на влакна (Fusion Splicer)**

Оптичното влакно се включва към жичната компютърна мрежа по няколко начина. Първият начин е чрез преобразувател (Media Convertor), който има един оптичен порт с два извода за влакна – приемник и предавател и един порт за усукана двойка с RJ-45 куплунг. На пазара има преобразуватели за различни скорости (100 Mbit/s, 1 Gbit/s) и различни разстояния (10 km, 30 km, 80 km). Примерен преобразувател е показан на фиг. 3.12.

**Фиг. 3.12. Преобразувател за оптично влакно.**

Съществуват и комутатори (switch) с един или няколко оптични порта и повече (8, 16, 24) порта за усукана двойка, към които могат да се включват компютрите на мрежата. Пример за такъв комутатор и SFP модул за връзка на оптично влакно е показан на фиг. 3.13.


**Фиг. 3.13. Комутатор с оптични портове и SFP модул.**

### 3.1.4 Безжични среди

Едно от доста бързо развиващите се направления в съвременните компютърни мрежи са безжичните компютърни мрежи. Те използват радиоефира като среда за предаване на информация, като излъчват радиовълни. Предимствата им пред останалите видове са, че при тях не се налага полагането на кабелни трасета, а компютърните системи могат да използват мрежата „по въздуха“. В повечето случаи обаче при тях разстоянията между компютрите са по-малки, от тези при кабелните мрежи и скоростите на обмен са по-ниски.

Съществуват няколко технологии за безжични мрежи, включително Bluetooth, Wi-Max и сателитните компютърни мрежи. Предмет на настоящата тема е групата технологии, стандартизирани в серията IEEE 802.11 и познати в практиката като Wi-Fi (Wireless Fidelity). Стандартите от фамилията, определящи честотите и начина на излъчване на сигнал в ефира са показани в таблица 3.2.

**Табл. 3.2 Стандарти за безжични мрежи**

| Стандарт | Ратифи-циран | Честотен диапазон | Теоретична максимална скорост | Разстояние при пряка видимост | Разстояние в сгради |
|---|---|---|---|---|---|
| IEEE 802.11 | 1997 | 2,4 GHz | 2 Mbit/s | 100 m | 20 m |
| IEEE 802.11a | 1999 | 5 GHz | 54 Mbit/s | 120 m | 35 m |
| IEEE 802.11b | 1999 | 2,4 GHz | 11 Mbit/s | 120 m | 35 m |
| IEEE 802.11g | 2003 | 2,4 GHz | 54 Mbit/s | 140 m | 38 m |
| IEEE 802.11n | 2009 | 2,4 или 5 GHz | 300 Mbit/s | 250 m | 70 m |

Показаните в таблицата разстояния са примерни и могат силно да се различават при различни условия. Те са определени опитно при използване на вградените антени на устройствата. При използване на външни антени с различно усилване е възможно достигане на разстояния до няколко десетки километра.

Понятието „Теоретична максимална скорост“ използвано в таблицата е определено в съответните стандарти и означава максималната скорост от битове, която може да премине през безжичната среда. При всички стандарти, освен 802.11n практическата скорост на връзката е около 50 – 55% от теоретичната при идеални условия, например при стандартите с теоретична максимална скорост 54 Mbit/s практическата е около 19 – 22 Mbit/s и може силно да спадне при прегради (например стени) или смущения.

Най-новият засега стандарт IEEE 802.11n работи по технологията MIMO (Multiple In Multiple Out), което означава че всяко устройство има вградени няколко приемника и няколко предавателя, всеки от които има теоретична максимална скорост 54 (или около 50) Mbit/s. Така например ако едно устройство има вградени 3 предавателя и 3 приемника, то производителят обикновено изчислява скоростта му така: 3 предавателя по 50 Mbit/s + 3 приемника по 50 Mbit/s = 150 + 150 = 300 Mbit/s. Това изчисление е силно спекулативно, защото практическата скорост зависи от много фактори. Ако например отсрещното устройство има само един приемник, то връзката би могла да бъде с полезна скорост около и под 20 Mbit/s.

Друг фактор, определящ полезната скорост при безжичните мрежи е броят потребители. Ако към едно централно устройство (точка за достъп – Access Point или маршрутизатор – Router) са закачени няколко потребителя, то те си поделят полезната скорост на устройството, като се налага техните компютри да се изчакват един друг, за да предават данни. Така ако имаме мрежа по стандарта 802.11g с теоретична максимална скорост 54 Mbit/s и практическа около 20 Mbit/s и към нея са закачени десет потребителя, те ще си поделят тази скорост и за всеки ще има грубо по около 2 Mbit/s. Това поделяне разбира се зависи от много фактори, например от местоположението на потребителите, от приложенията, които използват и други фактори и може да не е равно разпределено.

Безжичните мрежи обикновено работят в топология, при която едно централно устройство - точка за достъп (Access Point) или маршрутизатор (Router) осигурява връзката към мрежата (например Интернет), а едно или повече клиентски устройства (компютри, таблети, телефони) се свързват към централното и така използват услугата му. Съществува и възможност две клиентски устройства (например два компютъра) да се свържат помежду си директно, без използването на централно устройство. Тази връзка обикновено е означена в настройките като Ad-hoc и често има ограничени възможности, например скорост на предаване на данни.

Тъй като предаването на данни при безжичните мрежи е ефирно и става с еднакви устройства, то е лесно прослушването на сигнала от странични потребители на безжичната мрежа. Затова за да се повиши сигурността на данните е препоръчително да се използва механизъм за защита с парола и криптиране на данните. Повечето съвременни устройства поддържат три механизма за криптиране:

**WEP** (Wired Equivalent Privacy) е най-старият и най-несигурен метод за защита. Съществуват разработени атаки срещу него, при които ключът може да бъде откраднат в рамките на минути, затова не се препоръчва да се използва, освен ако в мрежата има някое старо устройство, неподдържащо останалите методи.

**WPA** (Wi-Fi Protected Access) е междинно решение, което повишава сигурността на WEP, но поради използване на някои части от стария метод, все още е уязвимо до известна степен.

**WPA2** е най-съвременният и най-сигурен метод за защита в безжични мрежи. Когато е комбиниран с дълга и сложна парола, за сега няма разработени смислени практически методи за атака.

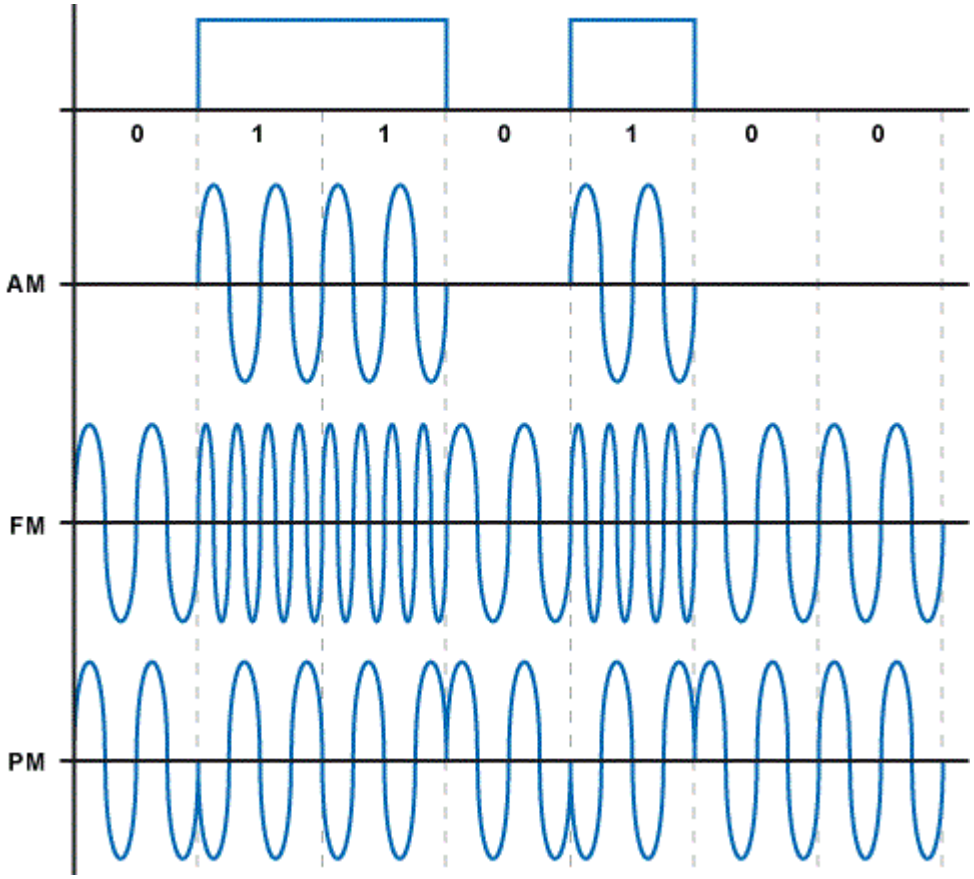
### 3.2 Сигнализация в компютърните мрежи

Сигналите се предават по средата под формата на електрически импулси при металните кабели, светлинни импулси при оптичните влакна или радиовълни при безжичните предавания. Те имат за цел да представят битовете цифрова информация (единици и нули), така че да могат да бъдат различени от приемника.

#### 3.2.1 Модулация

При предаването на цифров сигнал през аналогов канал за данни, често се налага той да се модулира, т.е. да се представи чрез по-висока честота, подходяща за пренасяне през канала. Съществуват три основни метода за модулация - амплитудна (Amplitude Modulation, AM), честотна (Frequency Modulation, FM) и фазова (Phase Modulation, PM). При амплитудната модулация единицата се предава с наличие на сигнал (висока амплитуда), а нулата – с липса на сигнал (ниска амплитуда). При честотната модулация има два сигнала – единият с по-ниска, а другия – с по-висока честота, като единицата се предава с по-високата, а нулата – с по-ниската честота. При фазовата модулация амплитудата и честотата на сигнала остават постоянни, а при промяна на предаваните битове (например от 1 в 0) се променя само фазата на сигнала, т.е. посоката на промяна на синусоидата – вместо да продължи да нараства, тя променя фазата си на $180^{o}$ и започва да спада.

На фиг. 3.14 е показан цифровия сигнал, състоящ се от поредица от единици и нули, при които единицата се представя с наличие на сигнал (обикновено +5V), а нулата – с липса на сигнал – 0V, а на долните три графики – примерно представяне на тази информация чрез амплитудна, честотна и фазова модулация.



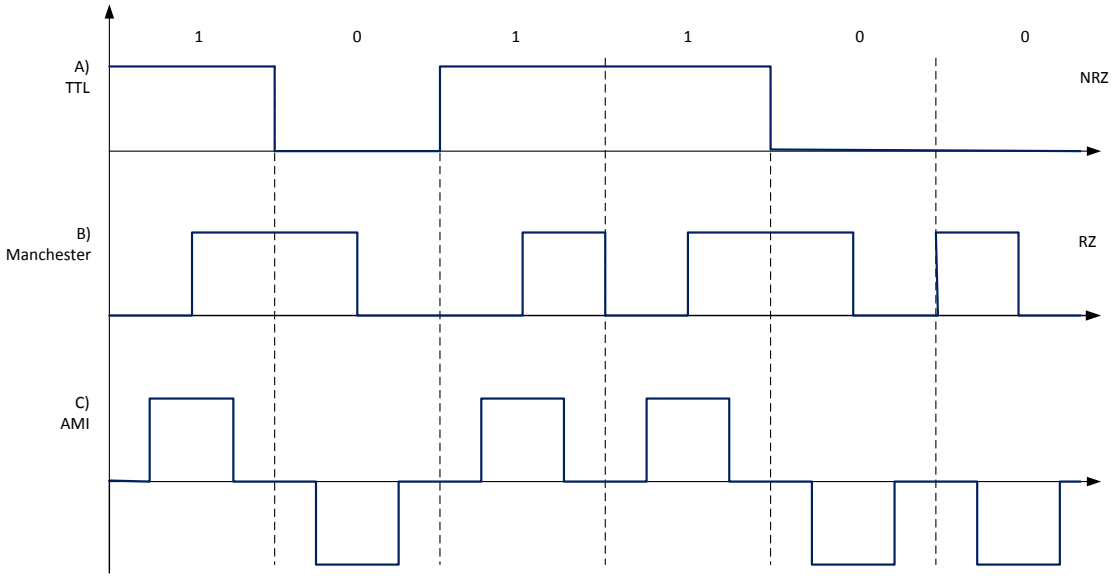

**Фиг. 3.14. Видове модулация.**

### 3.2.2 Сигнали с и без връщане в нулата

Представянето на цифровата информация в компютърните платки често се асоциира с така наречената TTL (Transistor–transistor logic) логика, при която единицата се представя с напрежение +5V, а нулата с напрежение 0V. Така интегралните схеми на компютъра различават единиците и нулите. Това е възможно благодарение на наличието на тактов генератор, който по отделна линия предава тактов сигнал, обозначаващ времето за предаване на отделните битове. Пример за предаване на единици и нули с TTL сигнал е представен на фигура 3.15 а. Този тип сигнали се наричат още NRZ (Non-return to Zero), тъй като при дълги поредици от единици сигналът никога не се връща в нулева стойност.

При компютърните мрежи обикновено средата е само една и ако се използва такъв вид кодировка може да се получи така, че при предаване на дълги поредици от единици или нули да се разсинхронизират тактовите генератори на предавателя и приемника и първият да предаде например 100 единици, а да бъдат приети 99. Затова представянето на сигналите в компютърните мрежи използва така наречените RZ (Return to Zero) кодове, при които сигналът във всеки бит се връща в нулата с цел синхронизация на тактовите генератори на предавателя и приемника. Пример за такъв код е Манчестърският код, използван при 10 Mbit/s Ethernet мрежите. Той е показан на

фигура 3.15 b и работи така: винаги в средата на предавания бит се прави преход на сигнала, което осигурява тактовата честота. Фронтът (посоката) на прехода определя каква стойност се предава – ако е положителен (от 0 в 1) се предава 1, ако е отрицателен (от 1 в 0) се предава 0. Недостатък на това представяне е двойното повишаване на честотата – за да се предаде сигнал със скорост 10 Mbit/s са необходими 20 MHz честотна лента.



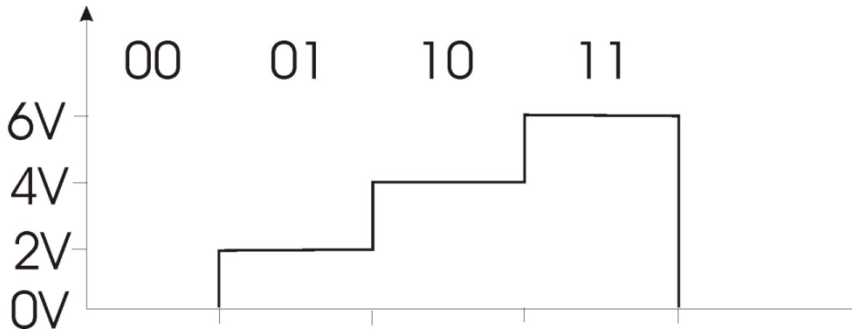

**Фиг. 3.15. Видове кодиране: a)TTL, b) Манчестърско, c) AMI**

На фигура 3.15 c е показан друг пример за сигнал с връщане в нулата, наречен AMI (Alternate Mask Inversion), при който единицата се предава с положителен импулс, а нулата – с отрицателен, а междинните сигнали с нулева стойност се използват за синхронизация.

### 3.2.3 Многонивово кодиране

При описаните до тук примери има две нива на кодиране на сигнала, например при амплитудна модулация +5V за единица и 0V за предаване на цифрова нула. Този начин на предаване не може да увеличава скоростта на данните над пределната честотна лента на средата за връзка. Съвременните предавания на данни имат за цел да увеличат колкото е възможно скоростта на предаване, един от възможните методи за постигане на това е многонивовото кодиране. Идеята му е да се използват повече от две нива за представянето на повече битове едновременно. Пример за многонивово кодиране е показан на фигура 3.16. На него вместо две нива – 0 и +5 волта са използвани 4 нива – 0, +2, +4 и +6 волта. Така за всеки такт имаме едно от четири възможни нива, към които можем да асоцииираме една от четирите възможни комбинации от два бита: 00, 01, 10 и 11. По този начин за един такт ще предаваме два бита информация. Ако вместо четири нива се използват осем нива, могат да се предават по три бита едновременно, при шестнадесет нива по четири бита и т.н. По

същия начин при другите модулации може да се използват повече от две честоти или повече от две изменения на фазата на сигнала.




**Фиг. 3.16. Многонивово кодиране**

Увеличаването на броя на нивата обаче не може да бъде безкрайно, тъй като то понижава шумоустойчивостта – разликите между отделните нива стават по-малки и евентуално външно смущение би могло по-лесно да повлияе на битовете, интерпретирани от приемника.

Съвременните предавания на данни за да увеличат скоростта на предаване често използват комбинация от няколко типа модулация (например амплитудна и фазова) и многонивово кодиране, позволявайки за един такт да се предават повече битове.

### 3.3 Режими на предаване на данни

Режимът на предаване на данни определя посоката, в която се предават данните и поредността на предаване. Съществуват три главни режима:

- Симплекс (еднопосочен) – при него данните се предават само в една посока – от предавателя към приемника;
- Полудуплекс (двупосочен с изчакване) – при него по средата могат да се предават данни и в двете посоки, но не едновременно, а само едната страна може да предава в даден момент от време;
- Пълен дуплекс (двупосочен едновременен) – при него могат да се предават данни и в двете посоки едновременно. Най-лесно постигането на пълен дуплексен канал се осъществява, комбинирайки две симплексни среди.

Обикновено оптичното влакно е симплексен канал за връзка, въпреки наличието на технологии за двупосочно предаване по едно влакно. Постигането на двупосочно предаване става чрез използване на две влакна – едно за предаване и второ за приемане. Безжичните мрежи по технологията Wi-Fi работят в полудуплексен режим – устройствата предават на една и съща честота, като се изчакват и само едно може да предава в даден момент от време.

# 4. Канално ниво

Каналното ниво има ключова позиция в компютърните системи, тъй като в него се намира границата между хардуера и софтуера за мрежова комуникация. То изпълнява доста важни функции.

### 4.1 Логическа топология на мрежата.

Различните мрежови топологии, описани в точка 1.3.4 имат различни характеристики, предимства и недостатъци. Както беше отбелязано компютърните мрежи имат физическа топология, която определя как физически са свързани компютрите с преносната среда и логическа топология, която определя как се чувстват компютрите спрямо останалите в мрежата. Физическата топология се определя от избраната технология за връзка на физическо ниво, а на канално ниво се определя логическата топология. В някои мрежи физическата и логическата топология съвпадат, в други се различават. Например при Wi-Fi мрежите физическата топология е звезда, защото има централен възел (точка за достъп или маршрутизатор) и около него са разположени отделните клиенти, но логическата топология е шина, тъй като всички устройства работят на една честота и само един може да предава за единица време.

### 4.2 Формиране на кадри.

Единицата за предаване на данни на канално ниво се нарича кадър (frame). Каналният протокол определя формата на кадъра, който се състои от определени по брой, функция, позиция и размер полета (fields). Повечето полета съдържат служебна информация за изпълнение на една функция на нивото. Примерен кадър, съдържащ повечето полета, типични за съвременните протоколи е показан на фигура 4.1.


**Фиг. 4.1. Примерни полета на кадър.**

Повечето канални протоколи имат полета, определящи началото и края на предавания кадър. Някои нямат специално поле за край на кадър, а разчитат на поле за дължина на кадъра. При протоколите, които позволяват повече възли в една мрежа (например Ethernet) има специално поле за адреси, чиято функционалност е по-добре описана в точка 4.5 – Адресиране. При някои протоколи, например PPP (Point-to-Point Protocol, протокол за двуточкови съединения) в една мрежа може да има само две устройства – предавателят и приемникът, но въпреки това и техните кадри имат поле за адреси. В някои реализации то не се използва, в други се използва за различни от адресирането цели.

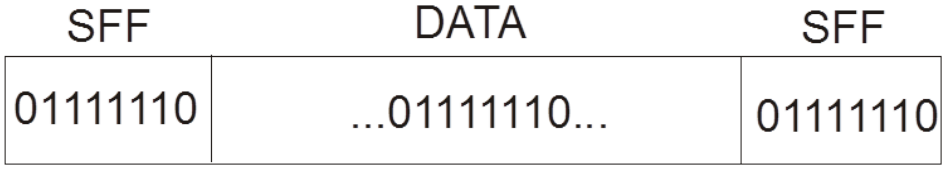
Някои канални протоколи имат различни видове кадри – управляващи, които се използват за управление на мрежовите устройства и информационни, които пренасят потребителските данни. Такива протоколи имат нужда от поле за тип на кадъра. При някои дължината на кадъра е фиксирана, а при други е променлива. Някои протоколи определят дължината на предаваните данни в заглавната част.

Всички канални протоколи имат поле за данни, в което опаковат информацията от по-горното мрежово ниво.

Повечето канални протоколи изпълняват функция за проверка за грешки. Тя е описана по-подробно в точка 4.4. Обикновено тази функция се изпълнява така: предавателят използва някакъв алгоритъм за изчисляване на допълнителна информация за проверка за грешки и я записва в полето за проверка. Приемникът използва същата функция върху съдържанието на получения кадър, изчислява стойност за проверка и я сравнява с тази, която е получил от източника. При разлика приемникът маркира грешка. След откриване на грешка някои канални протоколи изпращат съобщение за повторно предаване на сгрешената информация, а други просто изхвърлят сгрешеното и разчитат на протоколите от по-високите нива за отстраняване на грешката.

### 4.3 Кодопрозрачност.

Както вече беше споменато, в кадрите има някои служебни символи, означаващи например край на кадър. Нека да си представим, че символите за начало и край на кадъра са еднакви и са представени с битовата поредица 01111110. Какво би се случило, ако тази поредица се срещне в рамките на потребителските данни, както е показано на фигура 4.2? Вероятно приемникът би решил, че е дошъл края на данните и би прекратил приемането.



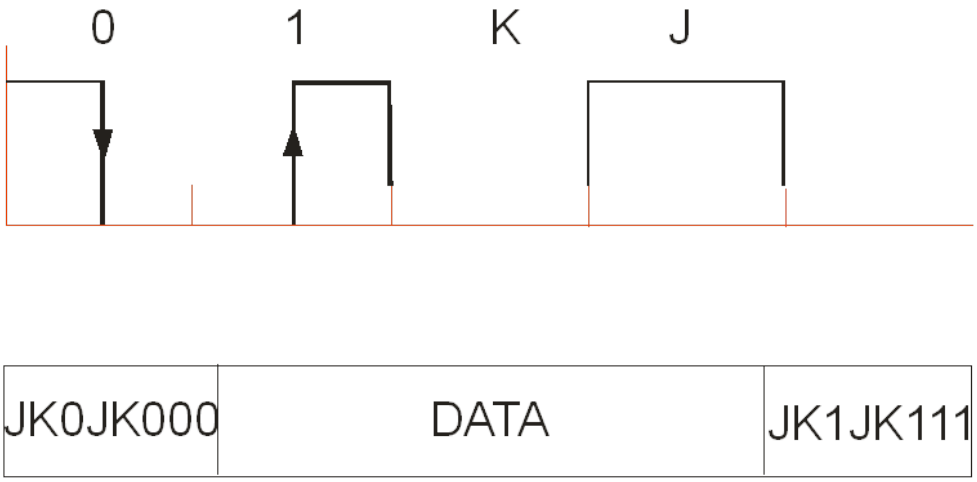

**Фиг. 4.2. Необходимост от кодопрозрачност.**

Функцията кодопрозрачност има за цел да осигури уникалност на служебните битови поредици, като позволи в рамките на данните да се срещат произволни битове, но те да се интерпретират като данни, а не като служебни символи. Ето някои примери:

#### 4.3.1 Кодопрозрачност при Манчестърско кодиране (10 Mbit/s Ethernet)

Както беше описано в точка 3.2.2, при Манчестърското кодиране данните се предават, като винаги в средата на бита има преход. За да може да се осигурят битови поредици, които не могат да се срещнат никога в рамките на валидни данни са

дефинирани двата специални бита, наречени J и K, при които в средата на бита няма преход. Възможните битове и поредиците за начало и край на кадъра при този протокол за показани на фигура 4.3.




**Фиг. 4.3. Кодопрозрачност при Манчестърско кодиране.**

Тъй като всички битове в тялото на кадъра имат преход в средата на бита (са 1 или 0), то използването на специалните битове за начало и край на кадъра гарантира уникалността на служебните символи.

### 4.3.2 Кодопрозрачност при 100 Mbit/s Ethernet – 4b/5b

При стандарта 100Base-TX, един от най-разпространените варианти на 100 Mbit/s Ethernet мрежи за осигуряване на кодопрозрачност се използва кодировката четири бита в пет бита (4b/5b). Други видове мрежови технологии използват идентични начини за кодиране, като 5b/6b, 8b/10b, но принципът е един и същ.

Идеята на метода е всички възможни 16 на брой комбинации от по 4 бита да се представят като пет-битови поредици, както е показано на фиг. 4.4.

| 4 битова поредица | 5 битова поредица | 4 битова поредица | 5 битова поредица |
|---|---|---|---|
| 0000 | 11110 | 1000 | 10010 |
| 0001 | 01001 | 1001 | 10011 |
| 0010 | 10100 | 1010 | 10110 |
| 0011 | 10101 | 1011 | 10111 |
| 0100 | 01010 | 1100 | 11010 |
| 0101 | 01011 | 1101 | 11011 |
| 0110 | 01110 | 1110 | 11100 |
| 0111 | 01111 | 1111 | 11101 |

**Фиг. 4.4. Заместване на поредици в 4b/5b.**

По този начин се постигат два ефекта – първо от всички 32 възможни пет-битови комбинации в рамките на валидни данни могат да се срещнат само тези 16, показани в таблицата. Така останалите 16 могат да се използват за служебни символи, като начало и край на кадър, по този начин се осигурява кодопрозрачност. Вторият ефект е свързан с необходимостта от кодовете с връщане в нулата, описан в т. 3.2.2 – пет битовите поредици от таблицата са така подбрани, че при всички възможни комбинации не могат да се получат дълги поредици от единици или нули, което премахва необходимостта от използване на код с връщане в нулата.

### 4.4 Управление на грешки.

Функцията за управление на грешки при някои протоколи е реализирана, за да могат само да откриват наличието на грешки, докато други реализации позволяват дори коригиране на грешките, за да не се налага повторно предаване на данните. Всички алгоритми за управление на грешки разчитат на изработването на допълнителна информация от съдържанието на данните чрез прилагане на математически операции и добавянето на тази информация към самите данни.

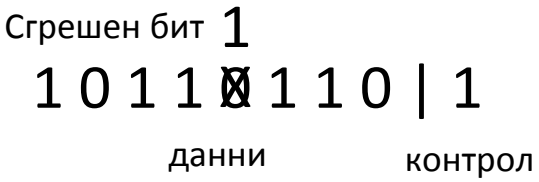
#### 4.4.1 Контрол по четност (нечетност)

Механизмът контрол по четност (Parity check) се използва за откриване на единични грешки при малки количества предавани данни. При него към предаваните данни се добавя един допълнителен контролен бит, чиято стойност е такава, че броят на всички единици в данните и контролния бит да е четно число (или нечетно число, ако използваме обратния механизъм – контрол по нечетност). Пример за контрол по четност е показан на фигура 4.5.

| Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 | контрол | Брой „1“ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 6 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 4 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 8 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 2 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Фиг. 4.5. Контрол по четност**

Към всяка порция от осем бита е добавен девети допълнителен бит, чиято стойност е такава, че броят единици в деветте бита да е четно число (нулата се счита за четно число). Данните се предават заедно с допълнителния бит. Приемникът брои единиците в приетите порции по девет бита и ако те са четно число, предполага че информацията е приета правилно. На фигура 4.6 е показан пример за единична грешка.




**Фиг. 4.6. Откриване на грешка.**

Ако при предаването единият бит се сгреши и се приеме като 1 вместо 0, то приемникът преброява единиците и открива, че те са 7 – нечетно число, така определя, че има грешка.

Методът не е идеален, защото не може да открие двойни (четен брой) грешки. Пример за такава е показан на фигура 4.7.


**Фиг. 4.7. Двойна грешка.**

В този случай при предаването са сгрешени два бита. Приемникът брои единиците, установява че те са четири – четно число и решава, че няма грешка.

Контролът по четност може само да открива грешки, но не може да разбере къде е самата грешка, затова не може да коригира, а само да сигнализира на предавателя за грешката и да изиска повторно предаване на сгрешената информация.

Този метод чето се използва при серийно предаване на данни към и от периферни устройства – клавиатури, мишки, принтери.

### 4.4.2 Блоков контрол по четност

Този алгоритъм е вариация на контрола по четност, като данните се разделят на блокове и им се прави контрол по четност и по редове и по колони, така че броят единици във всеки ред и във всяка колона заедно с контролния бит да е четно число. Пример за блоков контрол на блок от осем байта е показан на фиг. 4.8. Така ако един бит в блока е сгрешен, приемникът брои единиците във всеки ред и колона и получава нечетен брой в реда и колоната, където е грешката, както е показано на фигура 4.9. Така той не само разбира че има грешка, а разбира и къде е грешката, като в този случай може да я коригира, без да изисква повторно предаване на данните.

В случай на двойна грешка, например два последователни бита на един ред приемникът ще получи четен брой единици на всички редове, но на двете колони,

където е грешката ще има нечетен брой единици, т.е. той ще открие, че има две грешки, но няма да знае на кой ред точно са те и няма да може да ги коригира.

При този метод също е възможно да има грешки, които не могат да се открият, например ако се сгрешат четири бита, които се намират два по два в еднакви редове и колони, но все пак това е малко вероятна ситуация.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | |

**Фиг. 4.8. Блоков контрол по четност**

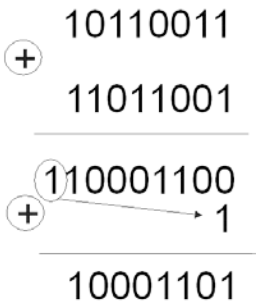
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | X | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | |

**Фиг. 4.9. Коригиране на грешка.**

Този метод се използва за корекция на грешки в някои компютърни памети, както и при някои технологии за глобални мрежи, при които данните се разделят на блокове при предаването.

### 4.4.3 Контролна сума (Checksum)

Контролната сума е лесен за изчисление метод, който с голяма вероятност открива грешки, но не може да ги коригира. Идеята му е да се разделят данните на равни по размер части (обикновено 16 или 32 бита), като първата се събира с втората. Ако при събирането се получи пренос (девети бит вдигнат в 1), то той се добавя към сумата. Резултатът от първите два се събира с третата порция и така до последната. Полученият краен резултат се добавя като контролна информация. Приемникът извършва същата операция с получените данни и сравнява резултата с изпратения от източника. Събирането на два байта с пренос е показано на фигура 4.10.




**Фиг. 4.10. Контролна сума.**

Методът се използва за проверка за грешки при някои мрежови протоколи, например TCP и IP.

### 4.4.4 Циклична проверка с излишък - CRC (Cyclic Redundancy Check)

Най-често използваният алгоритъм за проверка и коригиране на грешки в съвременните предавания на данни се нарича циклична проверка с излишък (CRC). Той се основава на теорията на полиномите от висока степен. Блокът предавани данни се представя като полином, който се разделя на предварително стандартизиран полином, обикновено от 16 или 32 степен и остатъкът от делението (16 или 32 бита) се записва на края на предаваните данни. Чрез повторно изчисление и сравнение с получените данни приемникът може да открие повечето грешки. При правилно избран делител се гарантира откриването на всички последователни грешки с дължина по-малка или равна на степента на полинома, всички последователни грешки с нечетен брой битове и с много голяма вероятност открива всички по-дълги от степента на полинома грешки с четен брой битове.

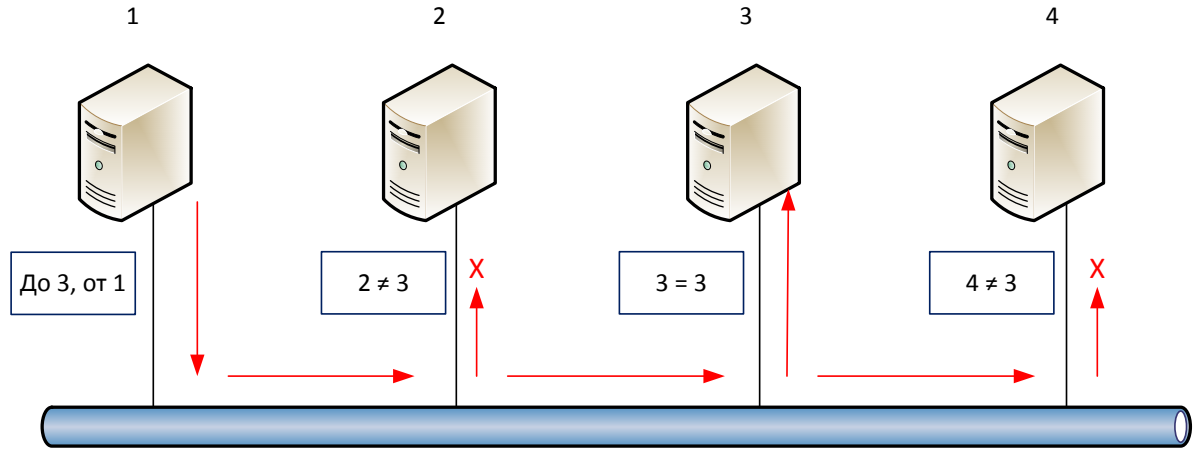
Някои реализации на CRC позволяват и да се коригират повечето единични грешки, но операциите, които трябва да се извършат в приемника често са прекалено сложни при голямо количество предадени данни, затова по-често методът се използва главно за откриване на грешки.

Мрежите от тип Ethernet използват полином от 32 степен и остатъкът е 32 битово число.

### 4.5 Адресиране.

Повечето мрежи с множествен достъп притежават функцията адресиране на канално ниво. Тя се осъществява чрез така наречените MAC (на български се чете мак) адреси, понякога наричани също хардуерни или физически адреси. В една мрежа с обща среда предаваният сигнал стига до всички получатели и ако няма адресиране на канално ниво всички ще трябва да приемат този сигнал, да го запаметят, да го проверят за грешки, да го обработят и по-късно да открият, че тези данни не са за тях. Казано с по-прости думи това означава всички компютри да приемат, запаметяват и обработват всички пакети, което е безсмислено разхищение на ресурси.

Използвайки функцията адресиране, преди да предаде данните в заглавната част на кадъра източникът записва двата адреса – своя като изпращач и на приемника като получател. Когато сигналът стигне до някоя станция, тя сравнява своя MAC адрес с този на получателя в кадъра и ако той не съвпада, станцията спира да приема тези данни, без да се налага да ги запаметява, проверява за грешки и обработва. Само станцията, която открие съвпадение на своя адрес със записания извършва тези дейности. Така всеки приема и обработва само данните, които са изпратени до него, както е показано на фиг. 4.11.

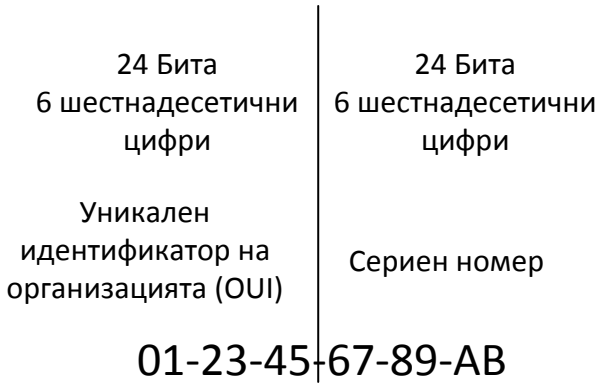



**Фиг. 4.11. Използване на MAC адрес**

В Ethernet мрежите MAC адресите са 48 бита. Прието е да се изразяват с шестнадесетични цифри – със символите 0 - 9 и латинските букви A – F. Буквата A изразява стойност 10, B – 11 и така нататък до буквата F, изразяваща стойност 15 – най-голямата стойност, която може да се съдържа във всеки разряд на шестнадесетичното число. Всяка шестнадесетична цифра представя четири бита, следователно MAC адресът се състои от дванадесет шестнадесетични цифри.

MAC адресите са проектирани така, че да бъдат уникални – лявата част на адреса определя организацията, която произвежда даденото устройство (компютър, мрежов

контролер, телефон), а дясната част представлява сериен номер на устройството. В съвременния свят при неспазване на правилата е възможно дублиране на MAC адреси, но тъй като те имат локално значение – определят изпращача и получателя между компютрите, споделящи обща среда за предаване, може да се получат проблеми в мрежовата комуникация само ако две устройства с еднакъв MAC адрес се намират в една и съща локална мрежа. Лявата част на адреса се нарича OUI (Organization Unique Identifier, уникален идентификатор на организацията) и се разпределя от IEEE срещу заплащане на такса от организациите, производители на хардуер. Така по лявата част на адреса може да се определи производителя на устройството или мрежовия контролер. Ако компютърът има няколко мрежови интерфейса (например жичен и Wi-Fi), те имат различни MAC адреси. Структурата на MAC адресите е показана на фиг. 4.12.




**Фиг. 4.12. Структура на MAC адрес.**

Съществуват няколко приети представяния на MAC адресите – дванадесетте шестнадесетични цифри да са групирани в шест двойки или в три четворки, като разделител може да се използва тире, точка или двоеточие, както е показано на фиг. 4.13. Независимо от представянето, на фигурата и трите изписвания изразяват един и същ MAC адрес.

01-23-45-67-89-AB
01:23:45:67:89:AB
0123.4567.89AB

**Фиг. 4.13. Примери за представяне на MAC адреси**

Съществува един специален MAC адрес с доста важно значение в съвременните мрежи. Това е адресът, в който всички 48 бита са със стойност 1 или адрес, който се изписва с 12 цифри F (FF-FF-FF-FF-FF-FF). Този адрес се нарича Broadcast и когато предадем данни до него, всички устройства в дадената локална мрежа припознават в

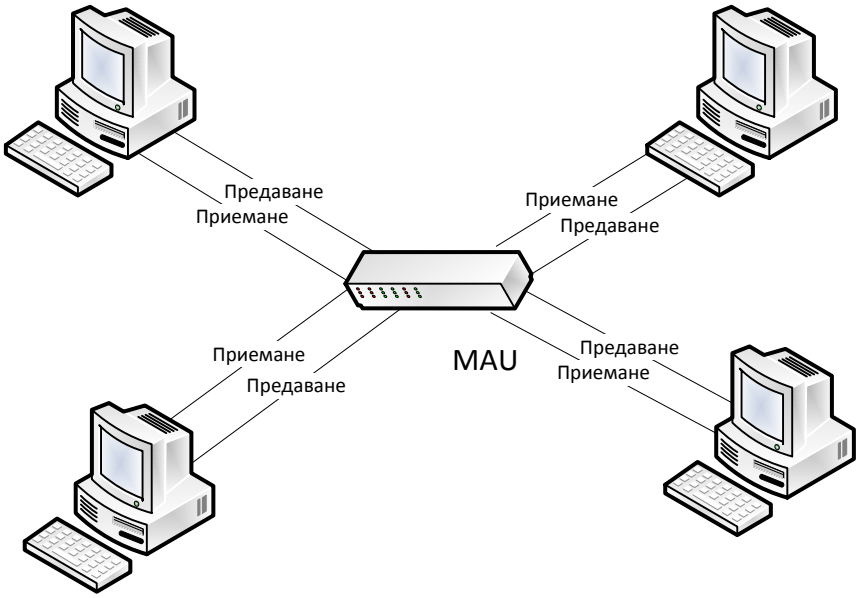
този адрес своя MAC адрес, приемат кадъра, буферират го и го обработват. Въпреки че това изглежда сравнително лесен начин да предадем едни и същи данни до всички в дадена мрежа, обикновено broadcast адресите се използват тогава, когато не знаем MAC адреса на получателя, за да предадем до всички и така да сме сигурни, че и желаният получател ще получи копие на данните. Това обаче означава, че всички останали обработват тези данни, а в крайна сметка те не са предназначени за тях. Затова broadcast предаването трябва да се използва внимателно и да се вземат мерки за неговото ограничаване. Някой протоколи, например ARP разгледан в следващите глави, използват такъв метод на предаване за да научат адресите на другите устройства в мрежата.

### 4.6 Управление на достъпа до средата.

Функцията „управление на достъпа до средата“ означава определяне на дисциплина на достъп – кой кога има право да предава данни. Обикновено дисциплините се делят на определени – при които е предварително определен начинът на разделяне на предаването (например по време) и неопределени – при които устройствата се състезават за придобиване на право на предаване.

#### 4.6.1 Определени дисциплини – Token Ring

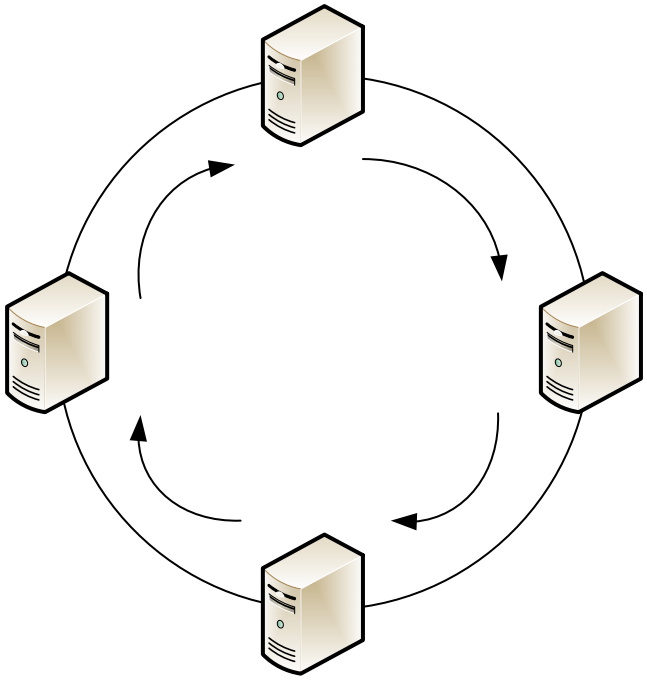
Най- популярния пример за определена дисциплина са мрежите от тип Token Ring (маркер по кръг), оригинално разработени от IBM и в последствие стандартизирани от IEEE в серията стандарти IEEE 802.5 за всички скорости на обмен, които са разработени и за Ethernet. Физическата топология на Token Ring е звезда и е показана на фигура 4.14.




**Фиг. 4.14. Физическа топология на Token Ring**

От централно устройство, наречено MAU (Multiple Access Unit) притежаващо няколко порта, по един кабел усукана двойка отива до всеки компютър и се включва в мрежовия му контролер.

Логическата топология обаче е кръг, поради факта че вътрешно в централното устройство приемната двойка на единия компютър се свързва с предавателната на предишния, а предавателната двойка – с приемната на следващия. Така всеки компютър може да приема сигнал само от предишния и да предава само на следващия, а последният – да предава на първия или компютрите са свързани в един логически кръг. Представяне на логическата топология е показано на фигура 4.15.



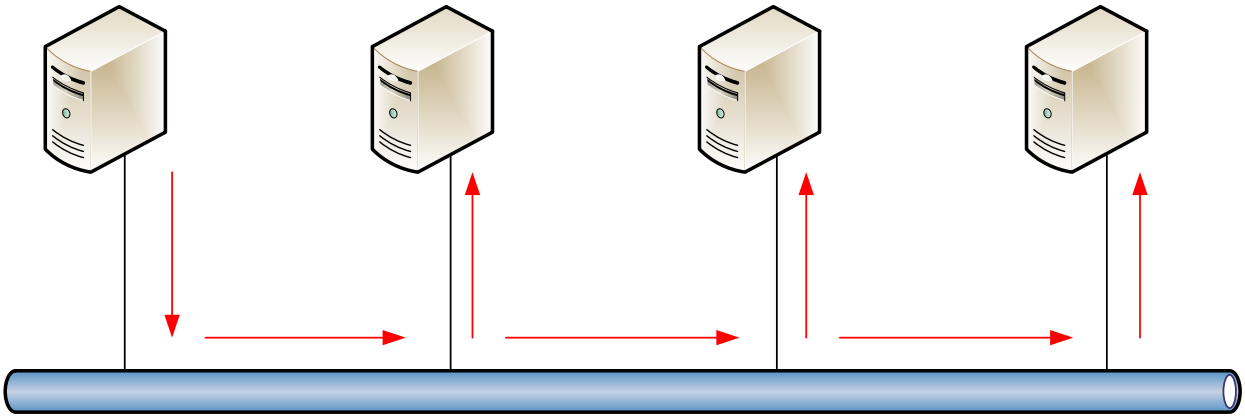
**Фиг. 4.15. Логическа топология на Token Ring**

Дисциплината на достъп е следната – в началото централното устройство генерира един специален кадър, наречен маркер (token) и го предава към първия компютър в логическия кръг. Когато даден компютър получи маркера има право да предава данни. Тогава той премахва маркера от кръга и предава данните, като ги означава с MAC адреси на получателя и предавателя. Когато данните се върнат пак при него, той разбира че те са получени, премахва ги от кръга и предава маркера към следващия. Ако компютър получи маркера и няма данни за предаване, той просто препредава маркера към следващия. Така в даден момент от време само един компютър предава данни, останалите трябва да чакат да дойде техния ред.

Мрежите от тип Token Ring са разработени като съвременни стандарти със скорости до 10 Gbit/s, но не са широко разпространени, затова няма да бъдат обсъждани повече в тази книга.

### 4.6.2 Неопределени дисциплини – Ethernet

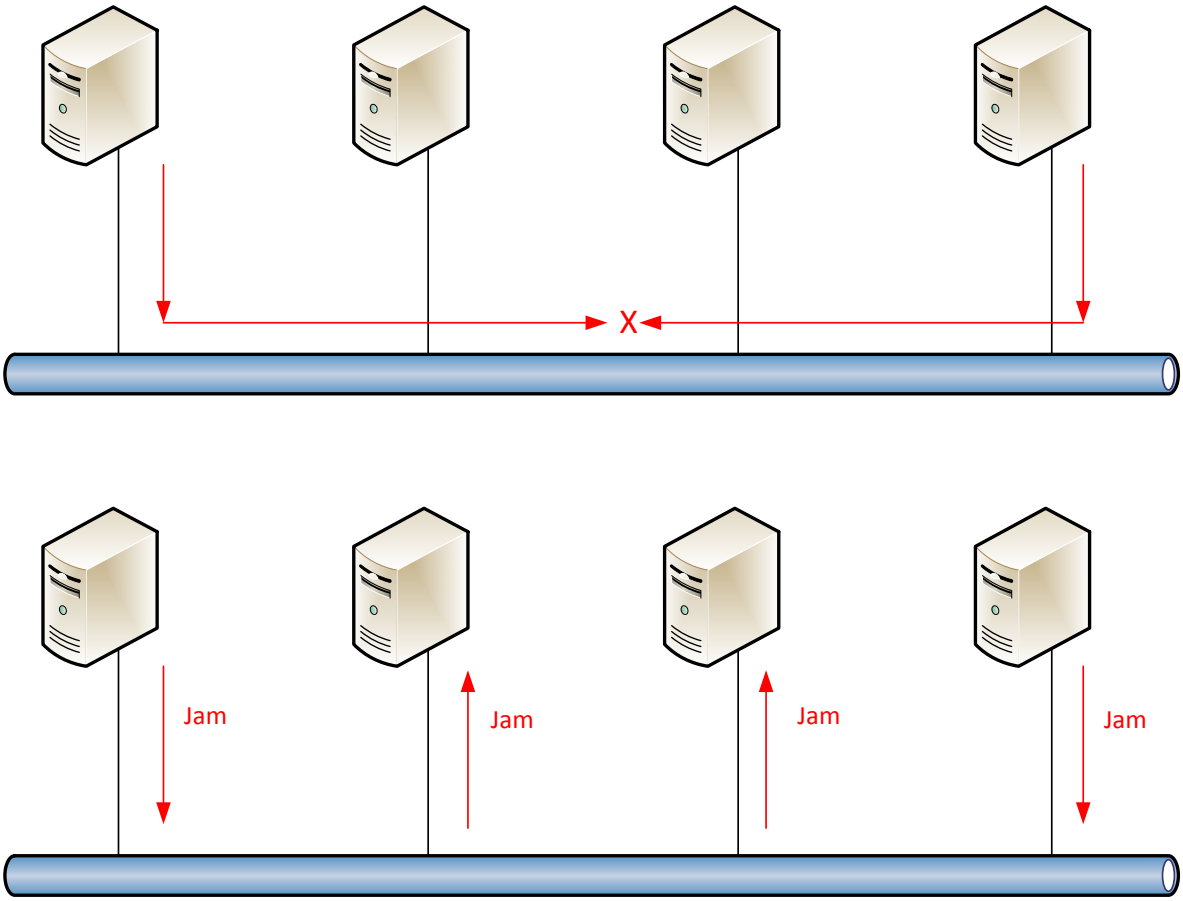
Безспорно най-разпространените и най-развиваните типове локални мрежи в световен мащаб в момента са мрежите от тип Ethernet. Първоначалните версии на този тип мрежи са разработени от Xerox през 1973 г. Дисциплината на достъп на тези мрежи се нарича CSMA/CD (Carrier Sense Multiple Access with Collision Detection). Дисциплината е неопределена, защото всички компютри, свързани към общата среда се състезават за заемането на средата. Когато един компютър иска да предава данни, той първо проверява дали е заета средата, т.е. дали някой друг предава. Ако средата е заета, той отлага предаването и след известно време опитва пак да предава. Ако средата е свободна, без да се чака някакъв сигнал се започва предаването. Така сигналът му се разпространява до всички и те сравняват MAC адреса на получателя със своя. Някои от старите реализации на този тип мрежа използват като физическа среда коаксиален кабел и физическата топология е шина. Други използват усукана двойка проводници и всеки компютър е свързан към централно устройство наречено концентратор (hub, хъб), т.е. физическата топология е звезда. Когато един компютър предава данни, сигналът стига до концентратора, той го усилва и го предава по приемните двойки на всички. Логическата топология и в двата случая е шина, защото само един може да предава за даден интервал от време, както е показано на фиг. 4.16.



**Фиг. 4.16. Логическа топология на Ethernet мрежи**

При тази дисциплина е възможно два или повече компютъра да проверят почти едновременно средата, да установят че тя е свободна и да започнат едновременно да предават данни. Тогава техните сигнали се наслагват и стават неразбираеми. Това явление е известно като колизия и е представено на фигура 4.17.

Всички компютри в мрежата трябва да разберат, че има колизия – тези, които предават трябва да разберат, че предаването е неуспешно и да опитат след време отново да предадат информацията. Тези, които приемат трябва да разберат че има колизия, за да изхвърлят приетите до момента невалидни данни и да очакват ново предаване.



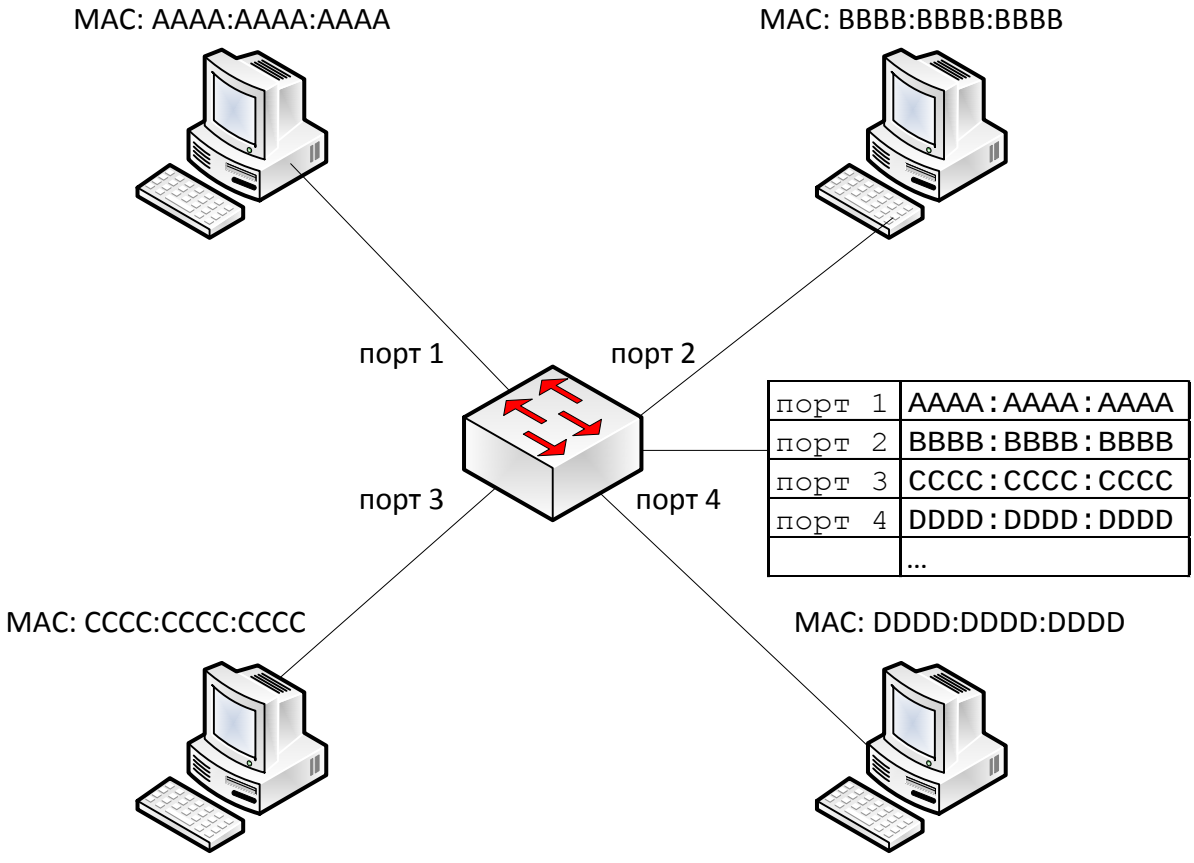

**Фиг. 4.17. Колизия и отстраняване.**

За тази цел предаващите компютри непрекъснато приемат сигнала по средата и го сравняват с този, който предават. Ако сигналът, разпространяващ се по средата се различава от предавания, то компютърът отчита колизия. Тогава той прекратява предаването на данни и започва да предава специалната битова поредица, наречена “Jam”, която не може да се срещне в рамките на валиден кадър. Така приемащите компютри, получавайки jam разбират, че приетият до момента сигнал е недействителен и изхвърлят текущо приетите данни. След като компютрите предизвикали колизията прекратят предаването и се изключат от средата, те чрез специален алгоритъм за псевдослучайни числа, наречен „backoff” изчисляват различно време, след изтичането на което се опитват отново да предадат информацията. Така компютрите работят в режим полудуплекс – или предават или приемат, не и двете едновременно.

Колизията е нормално явление в традиционните Ethernet мрежи, но реално времето, през което се получава, открива и отстранява колизията е загубено, понеже не се предават полезни данни. Затова е важно да се вземат мерки за намаляването и ограничаването на колизиите, доколкото е възможно.

### 4.6.3 Развитие на Ethernet.

В съвременните мрежи централното устройство е заменено с по-интелигентно – комутатор (switch, суич), който работи по различен начин, с цел по-ефективно предаване на данни и намаляване на колизиите. За постигане на тези цели комутаторът поддържа таблица, в която записва MAC адресите на компютрите в мрежата и портовете, на които те се намират, както е показано на фигура 4.18. Когато един компютър предава данни в заглавната част записва MAC адресът на получателя и своя MAC адрес за изпращач. Сигналът стига до комутатора и той вместо да го предаде навсякъде, както би направил концентраторът при старите реализации на Ethernet, той търси адреса в таблицата и ако го намери, предава само на този порт, където се намира получателя. Така когато се предават данни, те стигат само до получателя, а не до всички, което намалява натоварването на компютрите в мрежата. Освен това е възможно да се предават едновременно няколко сигнала, например компютърът на порт 1 да предава до този на порт 2 и едновременно с това компютърът на порт 3 да предава до този на порт 4, което повишава ефективността на мрежата.




**Фиг. 4.18. Ethernet мрежа с комутатор**

При тази дисциплина силно се намалява и броя на колизиите – ако няколко компютъра предават данни едновременно до един получател, комутаторът предава първия кадър и задържа втория в буфер, така че почти не се случват колизии. Това

означава, че вече не е необходимо когато даден компютър предава данни едновременно да слуша сигнала по средата. Затова в повечето случаи компютрите работят в режим пълен дуплекс – могат да предават до даден компютър по предавателната двойка и едновременно с това да приемат сигнал от друг компютър по приемната двойка. Всички тези подобрения спомагат за по-ефективната работа на съвременните мрежи.

### 4.7 Кадър на мрежи Ethernet.

Форматът на кадъра на мрежите от тип Ethernet е показан на фигура 4.19.

| Преамбюл 7 байта | Начало на кадър 1 байт | MAC адрес на получателя 6 байта | MAC адрес на изпращача 6 байта | Тип/ Дължина 2 байта | Данни 46 – 1500 байта | Проверка за грешки – CRC Код 4 байта |
|---|---|---|---|---|---|---|

**Фиг. 4.19. Формат на Ethernet кадъра**

Първите две полета са т.нар. преамбюл, който служи за синхронизиране на предавателя и приемниците и за сигнал за начало на кадъра. В следващите две полета се записват MAC адресите на получателя и изпращача. Полето тип/дължина има различно значение при старите и новите реализации на Ethernet – в началото там се е записвала дължината на кадъра. В настоящите реализации там се записва число, указващо протокола (например IP), чиито данни се пренасят в този кадър. Следва полето за данни. Минималната дължина на един валиден Ethernet кадър, който не е участвал в колизии е 64 байта, което означава, че към 18-те байта на полетата без преамбюла и начало на кадъра трябва да се добавят минимум още 46 байта или това е минималната валидна дължина на полето за данни. Ако станцията няма толкова данни за предаване, тя ги допълва с нули да 46 байта. Максималната дължина на стандартния Ethernet кадър е 1518 байта или максимум 1500 байта данни с добавени 18 байта за другите полета.

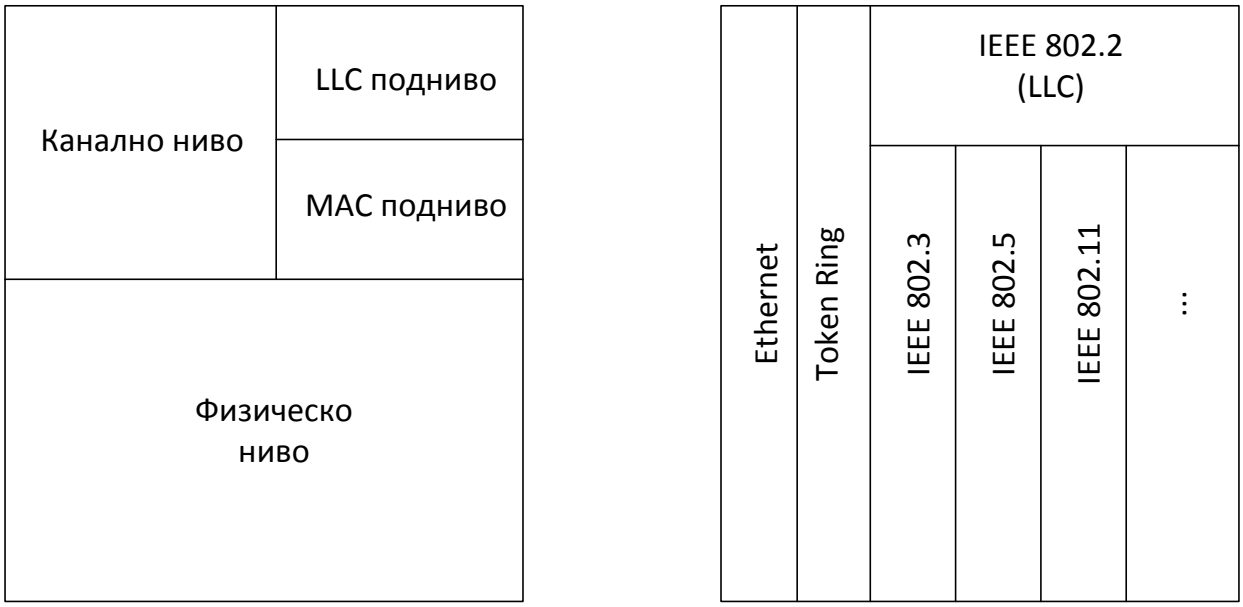
Кадърът завършва с 32 бита (или 4 байта) CRC код за проверка за грешки. При Ethernet мрежите няма изпращане на потвърждения. Когато приемникът изчисли CRC кода и установи че има грешка, той изхвърля информацията и не уведомява изпращача за грешката. Отстраняването на грешките е оставено на транспортното ниво.

### 4.8 Разделяне на каналното ниво на поднива.

Оригиналните реализации на различните видове мрежови технологии са разработени от различни фирми. Те често определят физическо и канално ниво – състоят се от преносна среда, съединители, мрежов контролер и драйвер за операционната система. В началните години на развитие на компютърните мрежи поради разликите в реализациите на технологиите и в начина им на взаимодействие с операционните системи често се е получавала несъвместимост между мрежови

технологии и версии на операционни системи. Например за да се премине на нова мрежова технология, да кажем Wi-Fi мрежа се налага да се закупи новата версия на дадена операционна система, която има реализирана поддръжка за тази технология.

За избягване на тези несъвместимости и за създаване на стандартно взаимодействие между мрежовите технологии и операционни системи в средата на 90-те години IEEE стандартизира разделянето на каналното ниво на две поднива. Горното ниво се нарича LLC (Logical Link Control, логическо управление на връзката) и описва стандартното взаимодействие между мрежова технология и операционна система. Долното ниво се нарича MAC (Media Access Control, управление на достъпа до средата) и определя логиката на съответната мрежова технология. Същевременно с това инженерите от IEEE пренаписаха и стандартизираха съществуващите по онова време мрежови технологии – Ethernet на Xerox беше стандартизиран като IEEE 802.3, Token Ring на IBM като стандарт IEEE 802.5 и други подобни. Всички тези нови стандарти вече взаимодействат със стандартното подниво IEEE 802.2, което взаимодейства по еднакъв начин с всички нови мрежови технологии и всички операционни системи. Схематично разделянето на каналното ниво на поднива е представено на фигура 4.20.




**Фиг. 4.20. Разделяне на каналното ниво на поднива.**

Съвременните мрежови технологии се разработват до MAC поднивото и взаимодействат със стандартното LLC подниво. Така разделянето практически означава, че всяка операционна система може да работи с всяка мрежова технология и разработката на нова технология не налага актуализация или смяна на операционната система.

# 5. Мрежово ниво – адресиране и функции на протокола IPv4

В момента на написване на тази книга, масовият протокол за достъп до Интернет е IPv4 (Интернет протокол, версия 4) и в тази част се разглеждат типичните за него мрежови параметри. Новият протокол, който се очаква да замести настоящия е вече готов, той се нарича IPv6 и е разгледан в следващата глава.

## 5.1 Структура на IPv4 адрес.

Всеки компютър, свързан към дадена мрежа трябва да има свой собствен уникален и глобално различим адрес в цялата мрежа. В предишната глава бяха разгледани MAC адресите и беше дефинирана тяхната роля – от всички компютри в една локална мрежа да се определи кой трябва да получи дадена порция информация. Но адресирането с MAC адреси разделя компютрите на производител и сериен номер. Няма логика, която да определя къде се намира даден компютър – дали в България или в друга страна. Затова в големи мрежи като Интернет е необходимо да се въведе друга схема на адресиране, която да има някаква степен на йерархия, определяща местоположението на компютрите.

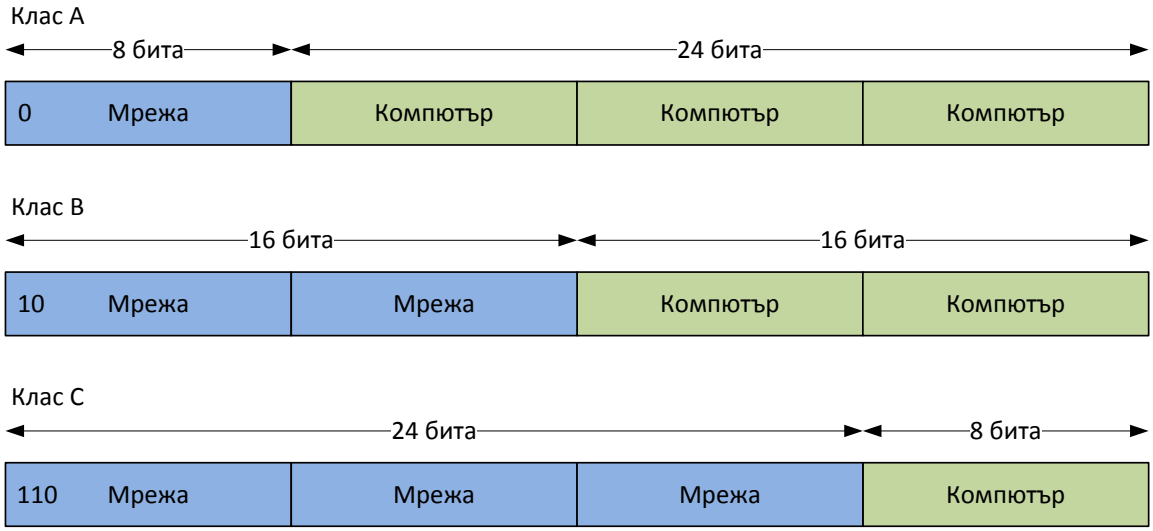
В протокола IPv4, текущо работещ в Интернет за адресиране се използват IP адреси. Те са 32 битови (4 байтови) числа, като по този начин е възможно да се адресират 4 294 967 296 компютъра. Поради йерархията на адресирането и за по-лесно възприемане е прието IP адресите да се изписват, като всеки от четирите им байта се представя с десетично число и между отделните байтове се поставя точка, например един възможен IP адрес е 192.168.1.5. Във всяка от цифрите възможните стойности са от 0 до 255, колкото са възможните стойности на един байт.

Йерархията на адресирането се осигурява по следния начин - лявата част на IP адреса означава номер на мрежата, в която се намира дадения компютър, а дясната част означава номер на компютъра в дадената мрежа. В различните IP адреси границата, разделяща адреса на мрежа и компютър е различна.

В началната реализация на протокола са дефинирани три класа адреси за достъп на компютри до Интернет – тези адреси се наричат още „комерсиални“, защото могат да бъдат раздавани на различни клиенти. Раздаването на адресите се дискутира малко по-късно в тази глава. Трите класа адреси са представени графично на фигура 5.1, а в таблица 5.1 са описани обхватът им, броят адреси и броят мрежи, които могат да бъдат достигнати във всеки клас.

За мрежите от клас A първият (най-левият) бит има стойност 0 – това определя класа на мрежата. За да познаем, че една мрежа е от клас A можем да погледнем най-лявата цифра на адреса. Понеже тя трябва да започва с бит 0, стойностите на цифрата могат да бъдат от 0 до 127. При тези мрежи първият байт е номер на мрежата, а останалите три байта – номера на компютъра в мрежата. Това са т. нар. „големи“

мрежи в Интернет. Както може да се види в таблица 5.1 – могат да съществуват 128 такива мрежи, като част от тях са заети за специални цели. Във всяка такава мрежа може да има над 16 и половина милиона компютъра.




**Фиг. 5.1. Класове адреси**

**Табл. 5.1. Параметри на класове мрежи.**

| Клас | Най-леви битове | Най-лява цифра | Брой мрежи | Брой адреси |
|---|---|---|---|---|
| A | 0 | 0-127 | 128 | 16 777 216 |
| B | 10 | 128-191 | 16 384 | 65 536 |
| C | 110 | 192-223 | 2 097 152 | 256 |

Мрежите от клас B са средните по размер мрежи и те започват с най-леви битове 10 или най-лява цифра от 128 до 191. В Интернет могат да съществуват над 16 хиляди такива мрежи, а във всяка от тях да има по над 65 хиляди компютъра.

Най-малките мрежи са от клас C и те започват с най-леви битове 110 или най-ляв байт от 192 до 224. Възможни са над 2 милиона такива мрежи, с по 256 адреса всяка.

Съществуват и два специални класа адреси. Клас D започва с битове 1110 или с най лява цифра от 224 до 240. Това са така наречените многоцелеви адреси (Multicast), при които е възможно едно предаване на данни, например един канал на Интернет телевизия, да стига до много (няколко) получателя едновременно. Идеята е данните да не достигат до всички абонати, защото това може да претовари мрежата при много телевизионни канали, а само до тези абонати които искат да гледат дадения канал.

Последният клас адреси са клас E, започващи с битове 1111 или с най-лява цифра от 241 до 255. Те са резервирани за изследователски цели. Доста време се считаше, че

те могат да се използват за резерва за допълнително разширяване на Интернет, сега е ясно че това няма да се случи.

### 5.2 Получаване на IP адреси

Дадена организация може да получи IP адреси за достъпа си до Интернет по два начина. Домашните абонати и по-малките организации обикновено получават своите IP адреси от доставчика си на Интернет услуги. Интернет доставчиците са организации, които имат голямо количество IP адреси, с цел да ги раздават на клиентите си и да им осигуряват достъп до Интернет. Неудобството на този начин е, че при смяна на доставчика клиентът трябва да смени и своя IP адрес – да вземе нов адрес от пространството на новия доставчик. Докато за домашните клиенти, които нямат сървъри, осигуряващи услуги в Интернет това не е голям проблем, за фирмите които имат инсталирани web или e-mail сървъри промяната на IP адреса води до временно прекъсване на услугите на тези сървъри и това е неудобно.

Интернет доставчиците и големите организации заявяват и получават своите адреси от регистратори – организации регулиращи ресурсите в Интернет. За различните региони от света се грижат следните регистратори:

- Европа: RIPE NCC (Réseaux IP Européens Network Coordination Centre, www.ripe.net);
- Северна америка: ARIN (American Registry for Internet Numbers, www.arin.net);
- Латинска америка и карибите: LACNIC (Latin America and Caribbean Network Information Centre, www.lacnic.net);
- Азия и тихия океан: APNIC (Asia-Pacific Network Information Centre, www.apnic.net);
- Африка: AFRINIC (African Network Information Center, http://www.afrinic.net/).

В отделни държави могат да съществуват и локални регистратори.

В момента на писането на тази книга раздаването на IPv4 адреси е силно затруднено поради тяхното изчерпване. Например европейският регистратор раздава последната си клас A мрежа (/8 порция), а азиатският обяви изчерпването на IPv4 адресите си през май 2011 година. Бъдещите разширения на Интернет са възможни с адресите на протокола IPv6, който е описан в следващата глава.

### 5.3 Запазени адреси

Всяка мрежа, независимо от размера си има два запазени адреса. Първият запазен адрес е адресът, при който всички битове в полето за номер на компютър имат стойност нула. На практика това е първият възможен адрес от мрежата, за мрежа от клас C последният байт има стойност нула, за мрежа от клас B последните два байта, а за мрежа от клас A – последните три байта са нули, например:

10.0.0.0 – за мрежа от клас A;
172.16.0.0 – за клас B;
192.168.1.0 – за клас C.

Тези адреси не могат да се назначават на никой от компютрите в мрежата и до тях не могат да се изпращат пакети. Тези запазени адреси се използват за означаване на самата мрежа. Чрез тях маршрутизаторите намират пътя до мрежата в Интернет. Процесът е описан в точката „Маршрутизация“.

Вторият запазен адрес от мрежата е адресът, при който всички битове в полето за номер на компютър имат стойност единица. На практика това са последните възможни адреси от всяка мрежа и в показание примерни мрежи за трите класа тези адреси изглеждат така:

10.255.255.255 – за мрежа от клас A;
172.16.255.255 – за клас B;
192.168.1.255 – за клас C.

Тези адреси също не могат да се назначават на компютър в мрежата, но до тях могат да с изпращат пакети. Когато изпратим пакет до такъв адрес, той се получава от всички компютри, които са включени в конкретната мрежа. Затова той се нарича broadcast, по подобие на радио или телевизионните излъчвания, при които един предава и сигналът се приема от всички настроени на съответната честота приемници. Пълното му означение е Directed Broadcast (насочен), защото принципно е възможно този пакет да бъде изпратен от външна мрежа и да бъде насочен към всички компютри от дадената мрежа, въпреки че на практика пакетите до този адрес често се филтрират от маршрутизаторите, свързващи ни към Интернет, с цел защита от някои атаки.

**5.4 Мрежова маска (Network Mask).**

Друг параметър, свързан с мрежовото адресиране при протокола IPv4 е мрежовата маска. Това е също 32 битова стойност, която има единици в битовете, които в IP адреса означават номер на мрежа и нули в битовете, които в IP адреса означават номер на компютър. Например за описаните по-горе класове мрежи A, B и C маските по подразбиране са:

255.0.0.0 за клас A;
255.255.0.0 за клас B;
255.255.255.0 за клас C.

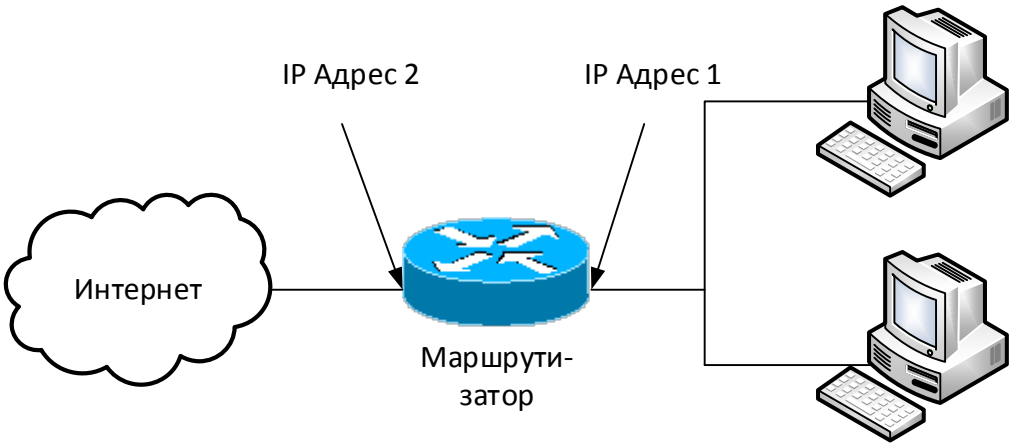
Това означава, че по подразбиране за мрежите от клас A първият байт (левите 8 бита) са номер на мрежа, останалите са номер на компютър, за клас B левите 16 бита са номер на мрежа, а за клас C – левите 24 бита. Мрежовата маска може да бъде различна, примери за това са показани в точката за разделяне на мрежа на подмрежи.

Съществува алтернативен начин за запис на мрежовата маска в комбинация с IP адреса на мрежата, наречен префикс. При него след IP адреса се поставя наклонена черта, след която се изписва броя битове от маската, които са единици. По-долу е показан записът на подразбиращите се маски за трите класа мрежи:

10.0.0.0/8 – за мрежа от клас A (съответства на 10.0.0.0 255.0.0.0);
172.16.0.0/16 – за клас B (съответства на 172.16.0.0 255.255.0.0);
192.168.1.0/24 – за клас C (съответства на 192.168.1.0 255.255.255.0).

### 5.5 Шлюз по подразбиране (Default Gateway).

Третият параметър за настройка при IP адресирането е шлюзът по подразбиране. Той има смисъл на връзка с всички други мрежи. Ако не настроим шлюз по подразбиране на някой компютър, той ще може да комуникира с останалите компютри в локалната мрежа, но няма да може да използва услуги от външни мрежи. На фигура 5.2 е показана типична локална мрежа, свързана към Интернет чрез маршрутизатор.

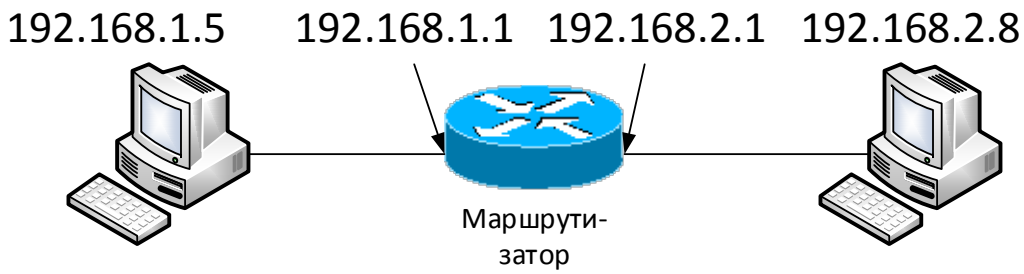



**Фиг. 5.2. Връзка към Интернет.**

За да могат компютрите от тази мрежа да използват Интернет, в полето Default Gateway на техните настройки трябва да се укаже IP адреса на маршрутизатора, който ги свързва към външния свят. Маршрутизаторът е устройство, което има поне два мрежови интерфейса, за да свърже поне две мрежи. Така на неговите интерфейси се поставят различни IP адреси – на вътрешния интерфейс трябва да се постави адрес от вътрешната мрежа, а на външния – адрес от мрежата на Интернет доставчика, към която сме свързани. За да изпълнява функцията си, в настройките на компютрите от вътрешната мрежа трябва да се укаже за шлюз по подразбиране вътрешният IP адрес на маршрутизатора. Така компютърът проверява IP адреса на получателя, до който трябва да изпрати пакета, вижда че той се намира в различна от неговата мрежа и го изпраща до MAC адреса на маршрутизатора, чиито IP адрес сме задали за шлюз по

подразбиране. Пуска пакета в мрежата и по MAC адреса той стига до маршрутизатора, който го приема и машрутизира – предава го към неговия получател.

Логиката, по която един компютър разбира дали неговия кореспондент е в неговата мрежа или в друга е следната: той умножава логически побитово двата IP адреса по мрежовата маска. Нека имаме две мрежи, свързани с маршрутизатор, както е показано на фигура 5.3.




```
  192.168.1.5 = 11000000.10101000.00000001.00000101
X
255.255.255.0 = 11111111.11111111.11111111.00000000
-----------------------------------------------------

  192.168.1.0 = 11000000.10101000.00000001.00000000


  192.168.2.8 = 11000000.10101000.00000010.00001000
X
255.255.255.0 = 11111111.11111111.11111111.00000000
-----------------------------------------------------

  192.168.2.0 = 11000000.10101000.00000010.00000000
```

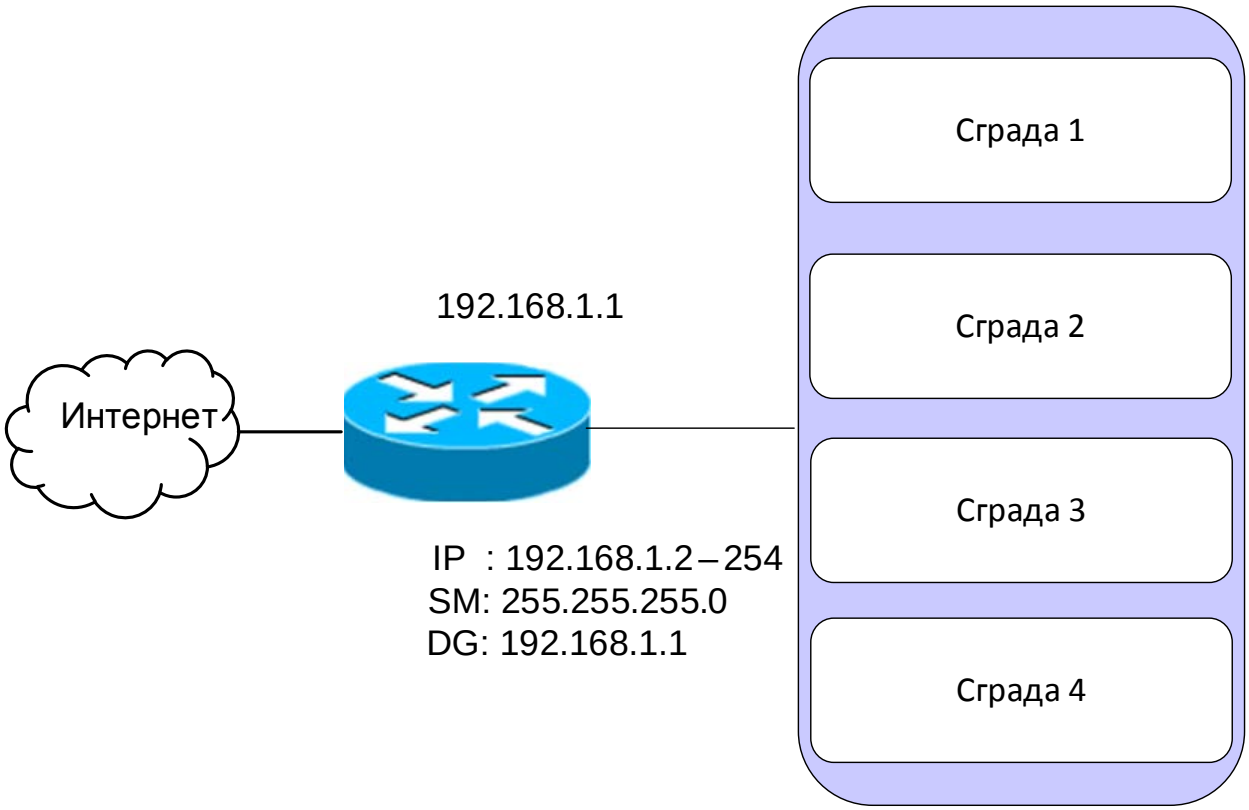
**Фиг. 5.3. Предаване през маршрутизатор.**

Ако компютърът отляво с IP адрес 192.168.1.5 трябва да предаде данни до компютъра вдясно с адрес 192.168.5.8, той умножава двата адреса по мрежовата маска. Понеже маската има единици, там където адресът означава номер на мрежа, тези полета остават непроменени, а нулите в полето за номер на компютър нулират тази част. Резултатът от умножението е запазения IP адрес, който има само нули в номера за компютър, т.е. адресът на мрежата. В примера резултатът означава – компютърът, който предава данни с адрес 192.168.1.5 е в мрежа 192.168.1.0, а получателят с адрес 192.168.2.8 е в мрежа 192.168.2.0. Така източникът разбира, че получателят е в друга мрежа и за да стигне до нея той трябва да предаде пакета на своя шлюз по подразбиране – маршрутизатора, който го свързва към другата мрежа. Действието на самия маршрутизатор е разгледано по-нататък в главата „Маршрутизация“.

### 5.6 Разделяне на мрежа на подмрежи.

Разделянето на мрежа на подмрежи има две цели. Първата е икономия на IP адреси. Ако сте Интернет доставчик и имате 64 клас С мрежи, които не можете да разделяте, независимо от големината на мрежите на клиентите си можете да им раздавате една или няколко цели мрежи с по 256 адреса. Така за домашните абонати, които имат по 1-2 компютъра се изразходват 256 адреса. Разделянето на мрежа на подмрежи позволява гъвкаво да се раздават на клиентите толкова адреси (кратни на степените на 2 – 2, 4, 8, 16,...), колкото са необходими и да не се губят излишно адреси.

Втората цел на разделянето на мрежа на подмрежи е филтрирането на излишен трафик. Представете си една мрежа, разположена в четири отделни сгради и свързана към Интернет, като показаната на фигура 5.4.

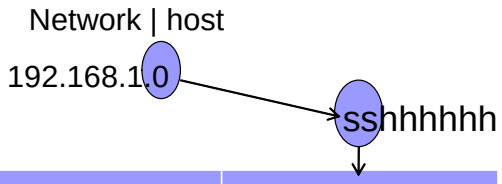



**Фиг. 5.4. Неразделена мрежа.**

В примерите е използвана мрежата от клас С 192.168.1.0, като първият възможен адрес 192.168.1.1 е назначен на маршрутизатора, а останалите незапазени адреси от 2 до 254 се раздават на компютрите в мрежата. Понеже няма логика как са разпределени адресите, то адрес 192.168.1.2 може да е във всяка една от сградите. За да работи такава мрежа, постъпил пакет от Интернет до които и да е от компютрите трябва да се изпрати във всички сгради. Ако някой от компютрите изпрати broadcast, той също ще достигне всички компютри и в четирите сгради.

За да намалим излишния трафик и да накараме пакетите да достигат само там, където е техния получател, а не навсякъде, трябва да разделим мрежата на подмрежи. Забележете, че разделянето има два компонента – физическо и логическо разделяне. По-добро от гледна точка на сигурност и функционалност решение е да се направи и физическо и логическо разделяне, но понякога от гледна точка на икономия на средства се прави само логическо, без физическо разделяне. Двата случая са разгледани по-нататък в примерите.

Нека първо да видим логическото разделяне, което определя кои IP адреси къде се намират и как устройствата разбират за начина на разделяне на мрежата. Принципът е показан на фигура 5.5.




| Мрежа | Broadcast | IP адреси | битове |
|---|---|---|---|
| 0 | 63 | 1- 62 | 00 |
| 64 | 127 | 65- 126 | 01 |
| 128 | 191 | 129- 190 | 10 |
| 192 | 255 | 193- 254 | 11 |

**Фиг. 5.5. Логическо разделяне на мрежа.**

Мрежата от клас C 192.168.1.0 в примера е разделена на четири подмрежи – по една за всяка сграда. Принципът на разделяне на мрежи дава възможност за разделяне на брой подмрежи, кратни на степените на две – 2, 4, 8, 16 и т.н. Ако по някаква причина ни трябват пет подмрежи за пет сгради, можем да разделим мрежата на осем подмрежи, като останалите три останат за резерва или в някоя от по-големите сгради назначим две или три подмрежи.

Разделянето става, като вземем няколко от битовете за номер на компютър и ги назначим за номер на подмрежа. В примера ни трябват четири подмрежи, затова взимаме два бита, с които образуваме $2^2$=4 комбинации. Ако вземем три бита ще имаме осем подмрежи, с четири бита можем да образуваме 16 подмрежи и т.н., но колкото повече подмрежи имаме, толкова по-малко IP адреси остават във всяка. Понеже IP адресът е йерархичен, трябва отляво да е номерът на мрежата, следван от номера на подмрежата и най-отдясно да е номерът на компютъра. Следователно трябва да вземем най-левите два бита, които стават номер на подмрежа, а останалите шест бита вдясно остават за номер на компютър (host) в съответната подмрежа. Техните четири възможни комбинации са показани в най-дясната колона от таблицата. В другите колони са показани номерата на новополучените подмрежи (образуват се,

като на всички най-десни шест бита за номер на компютър се даде стойност нула) и техните broadcast адреси (най-десните шест бита със стойност единици). Това са първият и последният адрес от дадената подмрежа. Всички стойности между тях са валидни IP адреси, които могат да се назначават на компютрите в съответната подмрежа.

В някои стари публикации може да се срещне твърдението, че при разделяне на мрежа на подмрежи, първата и последната подмрежа не бива да се използват. Причината е, че номерът на първата подмрежа (в примера 192.168.1.0) съвпада с номера на цялата мрежа, а broadcast адресът на последната подмрежа (в примера 192.168.1.255) съвпада с този на цялата мрежа, което може да причини известно объркване. Това отдавна вече не е валидна причина и в съвременния свят първата и последната подмрежи винаги се изполват, без това да причинява някакъв риск или объркване.

Как да кажем на устройствата в мрежата дали тя е разделена и как е разделена? Това става с промяната на мрежовата маска. Преди разделянето номерът на мрежата беше на границата на третия и четвъртия байт, затова маската беше 255.255.255.0 (или по съкратения начин на изписване /24). Сега към нея се добавят и двата бита, които взехме за подмрежа, следователно в новата маска левите два бита от последния байт трябва да станат единици или новата маска трябва да бъде 255.255.255.192 (/26).

Тъй като йерархията в структурата на IP адреса определя, че отляво е номерът на мрежата, а отдясно номерът на компютъра, то мрежовата маска винаги е непрекъсната поредица от единици, следвана от непрекъсната поредица от нули или в байт на мрежовата маска може да има една от деветте възможни стойности, показани на фигура 5.6.

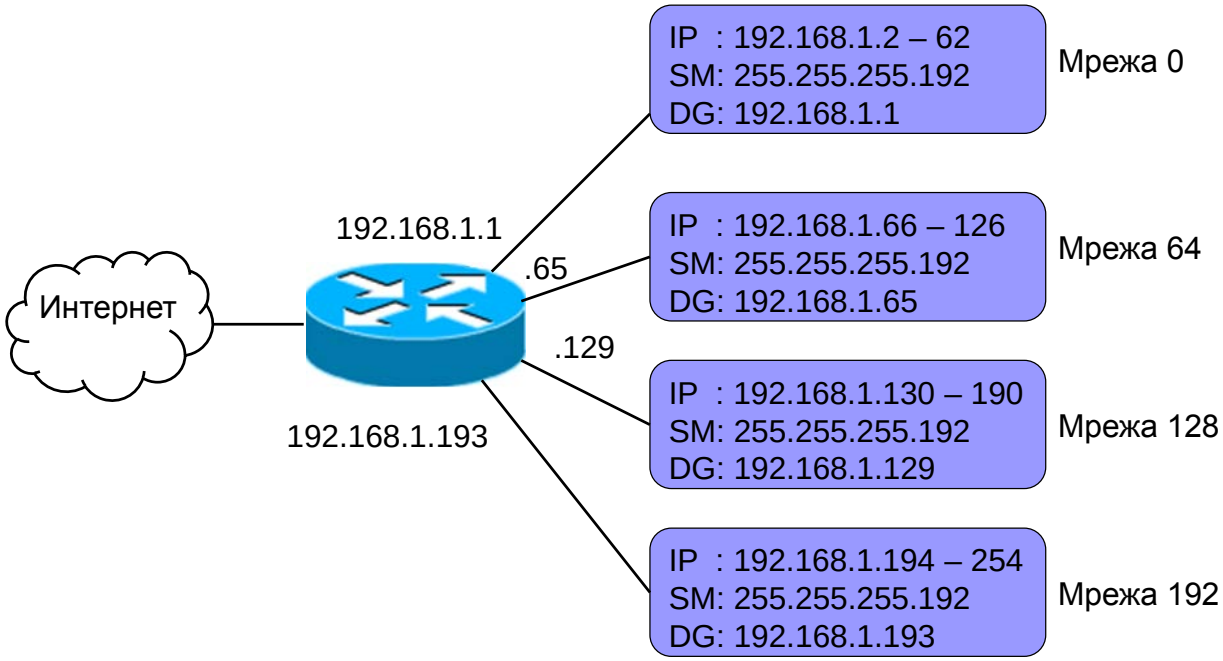
| | |
|---|---|
| 00000000 = 0 | 11111000 = 248 |
| 10000000 = 128 | 11111100 = 252 |
| 11000000 = 192 | 11111110 = 254 |
| 11100000 = 224 | 11111111 = 255 |
| 11110000 = 240 | |

**Фиг. 5.6. Валидни комбинации за байт на мрежова маска**

Разбира се, ако вляво от този байт на маската има други байтове, те трябва да са със стойност 255, а ако има такива вдясно, те трябва да са нули.

Сега когато вече имаме логическото разделяне на мрежата, можем да пристъпим и към физическото разделяне. Заменяме маршрутизатора с такъв, който има пет мрежови интерфейса – четири за всяка от сградите и един за връзка към Интернет. Прокарва се кабелно трасе за всяка от четирите сгради, заделя се по една подмрежа за

сграда, на всеки интерфейс на маршрутизатора се слага по един валиден IP адрес от всяка подмрежа и останалите адреси от подмрежата се дават на компютрите, разположени в съответната сграда. На всички интерфейси на маршрутизатора, както и на компютрите в съответната сграда се назначава новата мрежова маска (255.255.255.192) и за шлюз по подразбиране на всеки компютър се слага IP адреса на съответния интерфейс на маршрутизатора, свързващ подмрежата към останалите. Новата мрежа е показана схематично на фигура 5.7.

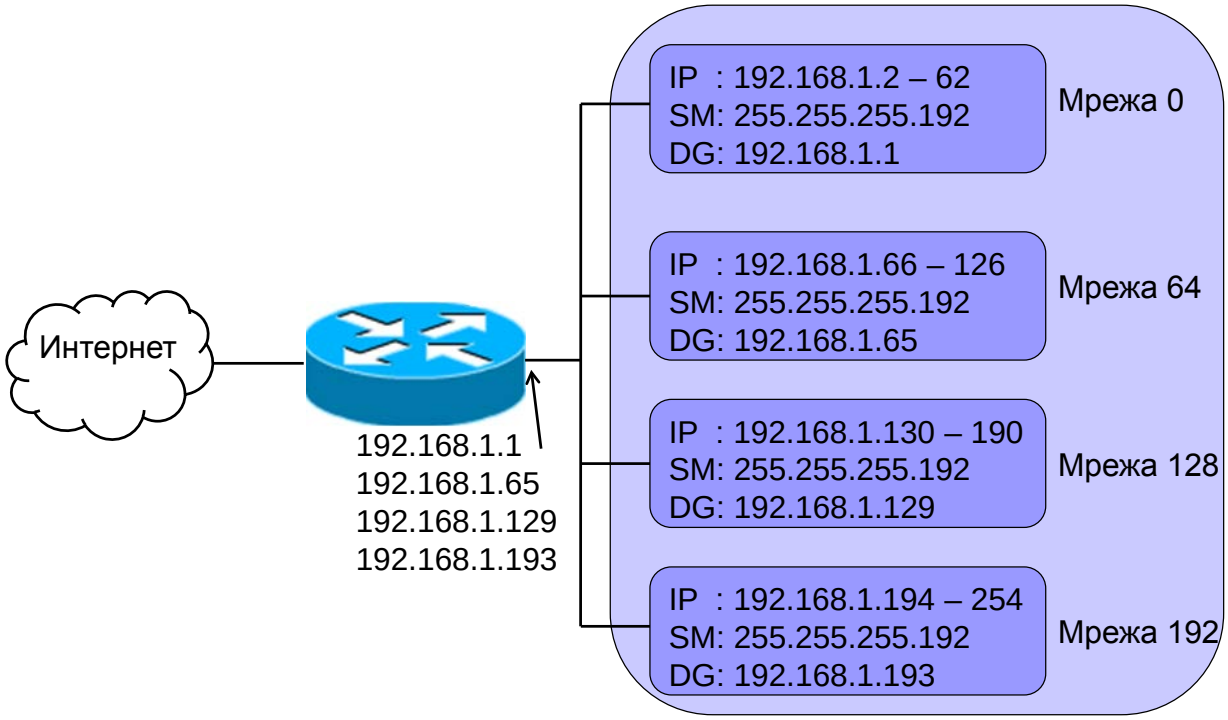



**Фиг. 5.7. Логическо и физическо разделяне на мрежа.**

Това е добър начин за разделяне на мрежата на подмрежи, при който физическата и логическата структура на мрежата съвпадат, но той изисква и повече средства – маршрутизатор с повече интерфейси и отделни кабелни трасета до всяка сграда. Понякога това е невъзможно да се осигури на практика, затова при някои по-икономични решения се пристъпва до логическо разделяне без физическо разделяне, както е показано на фигура 5.8.

В този пример кабелната система, свързваща сградите е една, машрутизаторът има един интерфейс свързващ мрежата към Интернет, но на този интерфейс са назначени едновременно четири IP адреса – по един за всяка подмрежа. Така ако компютър с адрес 192.168.1.100 иска да предава на компютър с адрес 192.168.1.200, той умножава двата адреса по мрежовата маска и установява, че източникът е в мрежа 192.168.1.64, а получателят в мрежа 192.168.1.192. Така следвайки логиката той предава на своя шлюз по подразбиране и въпреки че физически двата компютъра са в една мрежа те си комуникират през маршрутизатора. В него може да се настроят правила кой компютър с кои от другата мрежа има право да комуникира и така разделянето да помогне за

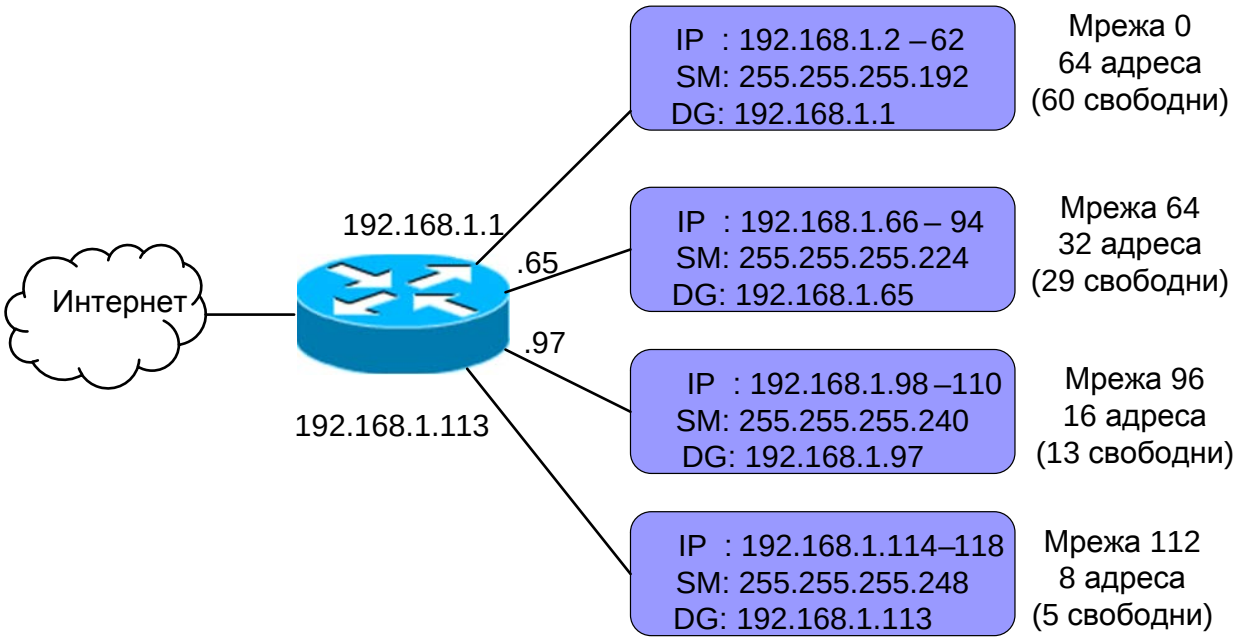
повишена сигурност. Това обаче е по-несигурният вариант, защото ако двамата кореспондента променят своите мрежови маски на оригиналната 255.255.255.0 техните компютри ще вярват че са в една мрежа и ще си комуникират директно, т.е. ще могат да преодолеят ограничението, поставено им при разделянето.




**Фиг. 5.8. Логическо разделяне на мрежа без физическо разделяне.**

В примерите за адреси на интерфейсите на маршрутизатора са избрани първите валидни адреси от всяка подмрежа. Това не е задължително, мрежата би работила с произволен адрес от достъпните за мрежата, стига той да е назначен за шлюз по подразбиране на компютрите, но е често прилагана практика на маршрутизатора да се задава първият валиден адрес от всяка мрежа.

Разгледаният пример показва разделяне на мрежа на равни по размер подмрежи, което може да е подходящо, когато има организация, разделена приблизително поравно в четири сгради, но не винаги е подходяща за доставчик на Интернет услуги, чиито клиенти са както домашни потребители с по един компютър, така и големи организации с по сто или повече компютъра. Затова в практиката съществува и разделяне на мрежа на различни по размер подмрежи, наричана още VLSM (Variable Length Subnet Mask), защото за различните по размер подмрежи маските са различни. Пример за такова разделяне е показан на фигура 5.9.




**Фиг. 5.9. Разделяне на различни по размер подмрежи.**

Както и в предишните примери броят свободни IP адреси със всяка мрежа е по-малък от кратния на степените на две размер на мрежата, защото във всяка подмрежа първия и последния адрес са запазени за номер на мрежа и Broadcast, а един от валидните адреси е назначен на маршрутизатора. Така в мрежа с 16 адреса можем да включим 13 компютъра, а в мрежа с 64 адреса – 61 компютъра.

### 5.7 Специални адреси.

Съществуват няколко IP адреса със специално предназначение, които не могат да се назначават на компютри за достъп до Интернет. Такива са:

- 0.0.0.0 – адресът с четири нули се нарича път по подразбиране (Default Route). За всеки компютър и маршрутизатор този адрес означава всичко, което не знаеш къде се намира изпращай натам, накъдето сочи този адрес. При компютрите обикновено той сочи към шлюза по подразбиране. При маршрутизаторите има по-специално значение, което е описано в главата за маршрутизация.
- 255.255.255.255 – нарича се локален broadcast (Local Broadcast). Чисто теоретично този адрес означава всички компютри в Интернет, но за предпазване от атаки всеки маршрутизатор спира пакетите изпратени до този адрес. Затова когато изпратим пакет до такъв адрес, пакетът стига до всички компютри от локалната мрежа, независимо какъв адрес има тя.
- Мрежа 127.0.0.0 или адрес 127.0.0.1 се нарича Local Host или Loopback. Когато даден компютър изпрати пакет до този адрес го получава той самия, без значение какъв друг IP адрес има назначен.

- Мрежата от клас B 169.254.0.0 е мрежа за автоматично IP адресиране. Когато някой компютър няма назначен IP адрес, той изпраща пакети чрез протокола DHCP до IP адрес 255.255.255.255, търсейки сървър, който да му назначи IP адрес и другите необходими параметри. Механизмът на протокола е разгледан детайлно в главата за приложно ниво. Когато обаче в мрежата няма работещ сървър за назначаване на адреси, компютърът си заема адрес от мрежа 169.254.0.0, изчислявайки останалите два байта от адреса на базата на своя MAC адрес. Тези адреси се наричат APIPA (Automatic Private IP Addressing) и компютрите с такива адреси нямат достъп до Интернет.

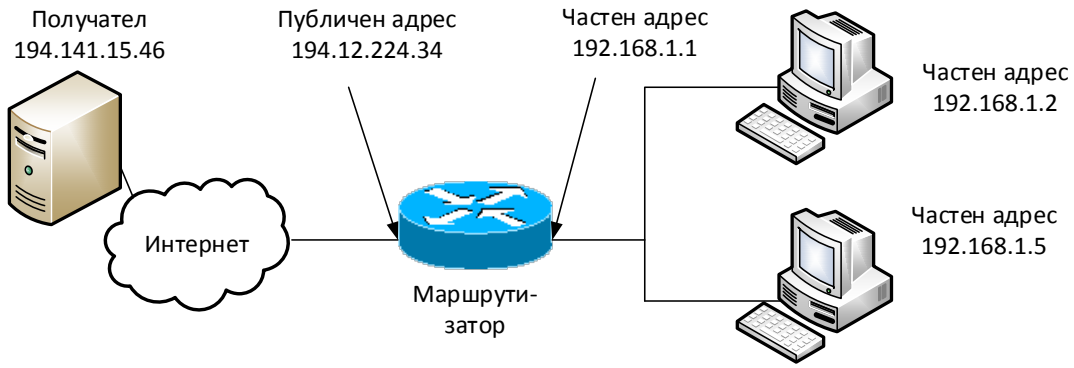
### 5.8 Частни IP адреси и превод на мрежови адреси (NAT).

Адресите от класове A, B и C без описаните специални адреси се наричат публични (public), защото компютър на който е назначен такъв адрес е публично достъпен от целия Интернет. Това налага практиката на разпределяне на IP адресите от регистратори. Ако трябва да се направи мрежа, работеща с IP протокол, но несвързана към Интернет не е необходимо да се искат адреси от регистратор или Интернет доставчик. Съществуват няколко мрежи, които са определени за частни адреси – адреси, които не са достъпни в Интернет. Всяка организация (от гледната точка на Интернет „частна“ означава несвързана към мрежата) може да вземе които и колкото желае такива мрежи и да назначи адресите на своите компютри. Тези мрежи са:

- Мрежата от клас A 10.0.0.0;
- Шестнадесет клас B мрежи от 172.16.0.0 до 172.31.0.0;
- 256 клас C мрежи от 192.168.0.0 до 192.168.255.0.

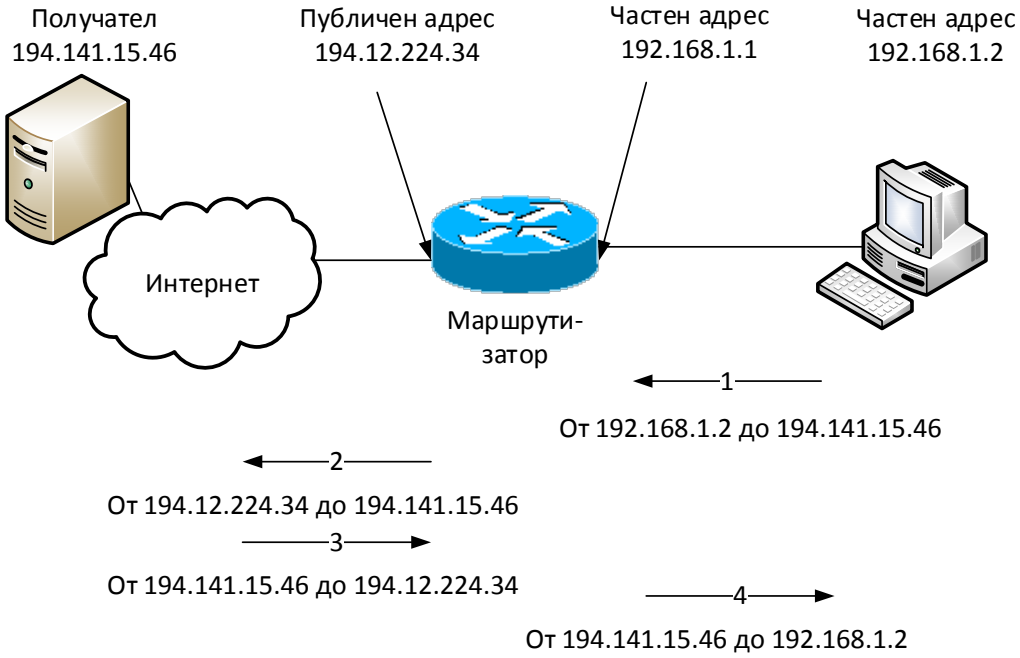
Използването на частни адреси намалява нуждата от публични адреси за устройствата в мрежата, които не трябва да са видими в Интернет. Понеже всеки маршрутизатор, свързан към Интернет изхвърля пакетите, изпратени до частен адрес, така се гарантира и по-високо ниво на сигурност – до компютър с частен адрес не могат да се изпращат пакети директно от Интернет.

Въпреки че оригинално компютрите с частни IP адреси са разработени да не могат да изпращат и получават пакети от Интернет, в съвременния свят е възможно компютри с такива адреси да използват Интернет, благодарените на техниката превод на мрежови адреси (Network Address Translation, NAT). Използвайки тази техника един или няколко компютъра във вътрешната мрежа, свързани към Интернет чрез маршрутизатор с включена NAT функционалност могат да получат Интернет достъп. Механизмът е показан на фигура 5.10.




**Фиг. 5.10. Техника NAT**

На фигура 5.11 са показани пакетите с адресите, както те се изпращат в посочения пример.




**Фиг. 5.11. Обмен на пакети при NAT**

Компютърът изпраща пакет от своя частен IP адрес, например 192.168.1.2 до публичния адрес на получателя – в примера 194.141.15.46. Понеже двата адреса са от различни мрежи, в заглавната част на Ethernet кадъра компютърът поставя MAC адреса на своя шлюз по подразбиране – този на маршрутизатора. Така пакетът стига до маршрутизатора. Ако той просто го препрати с частния адрес на източника, този пакет ще бъде филтриран. Затова преди да го предаде, маршрутизаторът изпълнява NAT – подменя частния адрес на компютъра 192.168.1.2 със своя публичен адрес – в примера 194.12.224.34 и така го изпраща в Интернет. Пакетът стига до получателя, той изпраща отговор от своя публичен адрес до публичния адрес на маршрутизатора, който получава пакета. На базата на информацията в своята таблица какви адреси е подменил, маршрутизаторът подменя отново своя публичен адрес с частния адрес на компютъра и предава пакета в локалната мрежа, така той достига до компютъра.

Благодарение на този механизъм компютри с частни IP адреси могат да използват Интернет. Въпреки че на втория пример е описан процес, при който има само един компютър с частен адрес, на базата на информацията, записана в табицата на маршрутизатора е възможно да се обслужват повече компютри с частни адреси, като всички те се подменят с един и същ публичен адрес – този на маршрутизатора. Някои производители, например Cisco Systems наричат този вариант PAT – Port Address Translation.

Техниката NAT също има известни проблеми, например ако искаме да инсталираме сървър на компютър с частен адрес, който да се вижда в Интернет, трябва да конфигурираме на маршрутизатора препращане на портове (Port Forwarding) – да укажем, че номера на порта, на който е инсталиран сървъра трябва да се препраща към частния IP адрес. Съществува и понятието „Демилитаризирана зона“ (DMZ), което означава всички пакети към публичния адрес на маршрутизатора да се препращат към един и същ компютър от вътрешната мрежа с частен адрес, без значение на номера на порта.

По-подробно описание за номерата на портовете, тяхното предназначение и разпределение е показано в главата „Транспортно ниво“. Необходимите номера на портове за различните програми и приложения и начините за конфигурирането на препращането на портове за много различни модели маршрутизатори е описано подробно със снимки в Интернет сайта: www.portforward.com.

### 5.9 Безкласово адресиране.

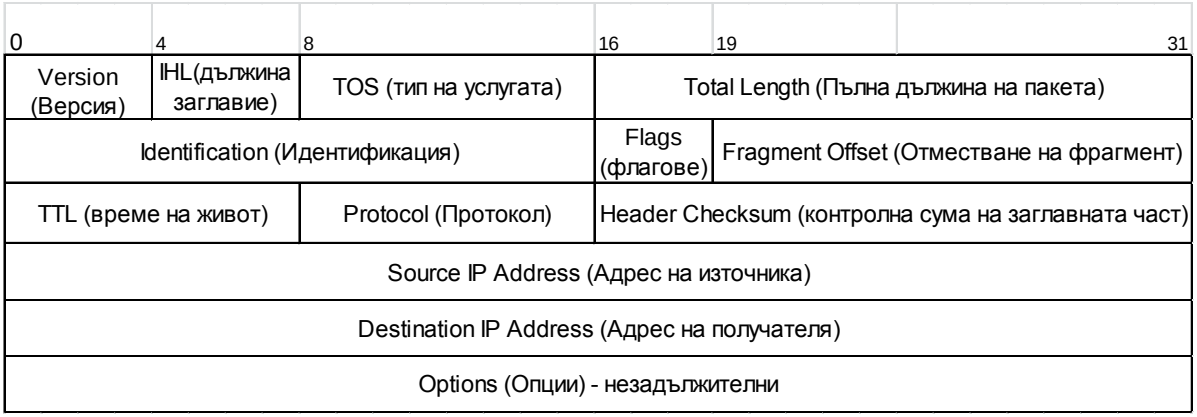
Класическото IP адресиране разделя адресите на класове и определя размера на мрежата според нейния клас – A, B или C. Но в последните години свободните IP адреси драстично намаляха, което наложи по-строги правила за тяхното раздаване. Това доведе до следния парадокс – имаше свободни мрежи от клас A с по почти 17 милиона свободни адреса, но никоя организация не покриваше изискванията да получи толкова адреси. Най-голяма част от организациите покриваха изискванията да получат една или няколко мрежи от клас C, но тези мрежи бързо се изчерпаха.

Идеята за безкласовото адресиране е описана за първи път през 1993 година, но практически навлиза масово след началото на новия век. Тя се основава на следния принцип – една свободна мрежа от клас A се разделя на 256 мрежи от клас B, всяка от тях се разделя на по 256 мрежи от клас C и така получените 65536 мрежи от клас C се раздават на организациите, които имат нужда от такива адресни пространства. По този начин се продължава животът на Интернет, но с цената на една важна промяна – устройствата вече не бива да съдят за размера на мрежата по нейния клас, а по мрежовата маска. Това налага промени в начина на работа на операционните системи и в протоколите, които се използват за маршрутизация – те вече трябва да предават помежду си мрежовата маска и да определят размера на мрежата спрямо нея.

Всички съвремени операционни системи са безкласови, но все още на света работят машини със стари операционни системи, работещи по класовия принцип. Това е текущото решение за продължаване на съществуването на Интернет, преди в него масово да навлезе новият протокол IPv6.

### 5.10 Заглавна част на IPv4 протокол.

Освен функцията адресиране, протоколът IPv4 изпълнява и други функции. Част от тях ще бъдат описани в следващите точки, но за да се разбере начинът на изпълнение на тези функции, тук ще бъде описан форматът на заглавната част на протокола с нейните полета. Тя е показана графично на фигура 5.12.




**Фиг. 5.12. Формат на заглавната част на протокола IPv4.**

На фигурата форматът на заглавната част е показан в 32 битови редове, като цифрите отгоре означават номерът на бита, от който започва съответното поле. Така се формират следните полета:

**Версия (Version)**: четири-битово поле, в поето се записва число, показващо номера на версията на IP протокола, който се използва. При IPv4 там пише числото 4, а при IPv6 – числото 6.

**Дължина на заглавие (Internet Header Length)**: четири-битово поле, в което се записва дължината на заглавната част в 32 битови порции. Тъй като в заглавната част на един IP пакет може да има или да няма опции (незадължителни допълнения), то минималната задължителна дължина на заглавната част без опции е 20 байта. За такъв пакет в полето се записва стойност 5 (дължината е 5 реда по 4 байта или 32 бита).

**Тип на услугата (Type of Service, TOS)**: Осем-битово поле, което се използва за различни приоритети на пакетите или качество на обслужването. По-подробно обяснение на полето има в следващите точки.

**Пълна дължина на пакета (Total Length)**: Шестнадесет-битово поле, в което се записва пълната дължина на пакета – заглавна част и данни в байтове. Това определя теоретичният максимален размер на един IP пакет – 65536 байта. В момента обаче

няма мрежова технология, която да може да пренася толкова големи пакети, затова при IPv4 е предвидена функцията фрагментация на пакетите, обяснена по-подробно в следваща точка. Фрагментацията се осигурява с помощта на следващите три полета – идентификация, флагове и отместване на фрагмента.

**Идентификация (Identification)**: Шестнадесет-битово поле, чиято стойност се установява от изпращача, за да помогне при асемблирането на фрагментите на една дейтаграма. Източникът, получателят, протоколът и идентификацията трябва да са уникални за всеки от изпратените пакети. Когато един пакет е фрагментиран, стойността от полето за идентификация на оригиналния пакет се копира и остава еднаква във всеки фрагмент.

**Отместване на фрагмента (Fragment Offset)**: Тринадесет-битово поле, което идентифицира местоположението на данните, носени от фрагмента спрямо началото на оригиналния, нефрагментиран пакет. Отместването се измерва в 8 байтови (64 битови) единици, т.е. за означаване на фрагмент, носещ данните на отместване 800 байта от оригиналния пакет, в полето се записва десетична стойност 100.

**Флагове (Flags)**: три бита, първият от които е запазен (не се използва). Другите са:

*Повече фрагменти (More Fragments, MF)*: еднобитово поле, което показва дали фрагментът е последният за пакета (при стойност 0) или има още фрагменти (при 1).

*Не фрагментирай (Don't Fragment, DF)*: еднобитово поле, което забранява (при стойност 1) на междинния възел да фрагментира пакета. При достигане на пакет с вдигнат бит DF до възел, който трябва да препрати информацията по трасе, което не може да предава толкова големи пакети, той не фрагментира пакета, а го изхвърля и връща съобщение за грешка на източника.

**Време на живот (TTL-Time To Live)**: осем-битово поле, което определя максималното време, измерено в секунди, за което пакетът може да остане в Интернет. Ако TTL съдържа стойност нула, пакетът трябва да бъде изхвърлен. Стойността на полето TTL се намалява във всеки междинен възел с единица за всяка започната секунда от движението на пакета, дори и ако времето, за което пакетът преминава до дадения възел е по-малко от една секунда. По този начин TTL е горната граница в секунди на времето за живот на пакета. Тъй като в съвременния свят повечето преходи в Интернет внасят времезакъснение по-малко от една секунда, на практика всеки междинен възел намалява това поле с 1, т.е. на практика то означава максималния брой преходи, които може да направи пакета между отделните мрежови възли, преди да бъде изхвърлен.

**Протокол (Protocol)**: в това поле се записва число, указващо протокола, чиито данни се пренасят от IP пакета. Например за протокол TCP стойността е 6, а за UDP – 17.

**Контролна сума на заглавната част (Header Checksum)**: В това шестнадесет битово поле изпращачът записва изчислената контролна сума на заглавната част на пакета по алгоритъма, показан в главата за канално ниво. Всеки междинен възел и крайният получател при приемането на пакета изчисляват контролната сума и сравняват получената стойност със записаната в полето, за да проверят за грешки в заглавната част. При наличие на грешка възелът изхвърля пакета и изпраща съобщение на източника за грешката. Тъй като всеки възел, през който минава пакета променя най-малко времето на живот, то преди да предаде пакета, той изчислява наново контролната сума и я записва в полето, за да може следващия възел да провери за грешки.

**Адрес на източника (Source IP Address)**: в това 32 битово поле се записва IP адресът на предаващия данните.

**Адрес на получателя (Destination IP Address)**: в това 32 битово поле се записва IP адресът на получателя на данните.

**Опции (Options)**: Незадължителни полета, които служат за допълнителни функции, рядко използвани в съвременния свят. Примери за опции са:

*Маршрутизация от източника (Source Routing)*, при която изпращащият пакета записва в заглавната част IP адресите, през които иска да премине неговия пакет.

*Запис на маршрута (Record Route)*, при която всеки възел по веригата добавя своя адрес в заглавната част, за да може да се проследи маршрута.

*Марка за време (Timestamp)*, при която всеки възел записва точното време според неговия часовник, когато пакетът е преминал през него.

### 5.11 Тип на услугата

Форматът на полето тип на услугата (Type of Service, ToS) е показан на фигура 5.13.

| | | | | | |
|---|---|---|---|---|---|
| Приоритет (Precedence) | Времезадръжка (Delay) | Скорост (Throughput) | Надеждност (Reliability) | Цена (Cost) | 0 |

**Фиг. 5.13. Поле „Тип на услугата“**

Първите три бита са приоритет, който е определен в следните осем възможни комбинации:

0 (000) – Обикновен пакет (Routine)
1 (001) – Приоритетен пакет (Priority)
2 (010) – Незабавна доставка (Immediate)
3 (011) – Светкавична доставка (Flash)

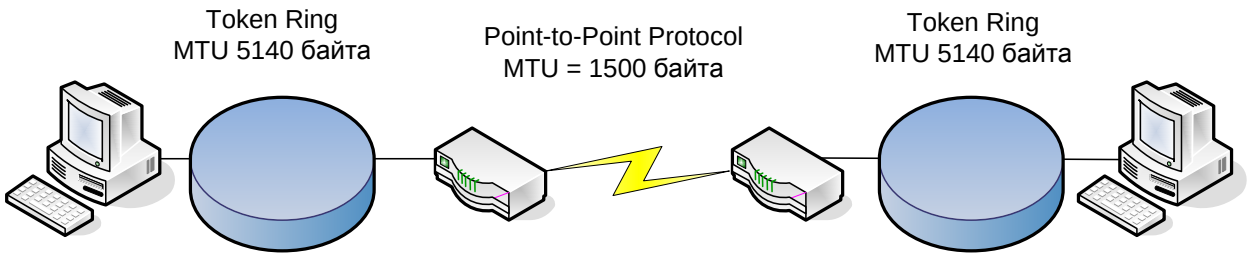
4 (100) – Повече от светкавична (Flash Override)
5 (101) – Критична (Critical)
6 (110) – Управление на междумрежовата връзка (Internetwork Control)
7 (111) – Управление на мрежата[1] (Network Control)

Следващите четири бита имат смисъл на „Оптимизирай за“: закъснение, скорост на трасето, надеждност на предаването и цена на преноса. Последният бит не се използва и има стойност нула.

Тези битове могат да се използват при осигуряване на функцията „Качество на обслужването (Quality of Service, QoS) и обикновено имат значение само в локалната мрежа на озганизацията, за да разделят един вид трафик от друг, например IP телефонията от файловия трансфер и да могат да обслужват едните с предимство спрямо другите. Обикновено тази функция не се използва в Интернет.

## 5.12 Фрагментация на пакети

Фрагментацията на пакети е функция, която е предвидена за съвместимост между различни програмни приложения и хардуерни мрежови реализации. Примерна постановка за фрагментация е показана на фигура 5.14.




**Фиг. 5.14. Примерна постановка за фрагментация.**

В примера две мрежи от тип Token Ring с максимален размер на пакета (Maximal Transmission Unit, MTU) 5140 байта са свързани чрез мрежа, в която максималният размер на пакета е 1500 байта. Понеже драйверът на мрежовия контролер на компютрите определя, че в локалната мрежа те могат да предават пакети с размер 5140 байта, техният софтуер се опитва да предава по толкова, за да използва оптимално мрежовата връзка. Когато пакетът достигне до маршрутизатора, той определя пътя за пакета и разбира, че мрежата в която трябва да предаде пакета може да предава максимално по 1500 байта наведнъж. За да осигури съвместимост, маршрутизаторът фрагментира пакета – разделя го на отделни фрагменти, всеки от които има своя IP заглавна част и пренася поредната порция от данните. Оригиналният

[1] Българските наименования на приоритетите са интерпретация на автора и не претендират за терминологична точност.

пакет с размер 5140 байта (5120 байта полезна информация + 20 байта заглавна част) се разделя на четири отделни фрагмента. Стойностите на полетата, свързани с фрагментацията за оригиналния пакет и получените фрагменти при тази ситуация са показани в таблица 5.2.

**Табл. 5.2. Фрагментация на пакет.**

| Identification | Total Length | DF | MF | Fragment Offset |
|---|---|---|---|---|
| 1234 | 5140 | 0 | 0 | 0 |

Оригинален пакет

| Identification | Total Length | DF | MF | Fragment Offset |
|---|---|---|---|---|
| 1234 | 1500 | 0 | 1 | 0 |
| 1234 | 1500 | 0 | 1 | 185 |
| 1234 | 1500 | 0 | 1 | 370 |
| 1234 | 700 | 0 | 0 | 555 |

Фрагменти

Полето Identification запазва стойността си при фрагментите, за да определи тяхната принадлежност към оригиналния пакет. Полето Total Length описва цялата дължина на фрагмента, заедно със заглавната му част (1500 байта фрагмент = 1480 байта данни + 20 байта заглавна част). Флагът More Fragments (MF) е единица за всички фрагменти без последния, където обозначава край на фрагментираната поредица. Полето Fragment Offset има стойност 0 в първия фрагмент, тъй като неговите данни се записват на отместване 0 в оригиналния пакет. За втория фрагмент стойността е 185, което означава, че неговите данни се записват на отместване 8*185 = 1480 байта и т.н.

Процесът на дефрагментация се състои в събирането на данните в цяла дейтаграма, според съдържанието на полетата Fragment Offset и Identification.

С годините фрагментацията е показала, че освен ползи може да донесе и доста проблеми – например понижена производителност, повишено натоварване на маршрутизаторите, податливост към грешки при предаванията и други. Затова съвременните препоръки са фрагментацията да бъде избягвана. В съвременните операционни системи има вграден алгоритъм за избягване на фрагментацията, наречен Path MTU Discovery (PMTUD). В новия протокол IPv6 мрежата вече не осъществява фрагментация.

# 6. Интернет протокол версия 6 (IPv6)

Протоколът, който се очаква да замести текущия IPv4 в Интернет се нарича IPv6. В някои публикации може да се срещне като „новият“ Интернет протокол или IPng (Internet Protocol new generation). Всъщност началната версия на този протокол е създадена през 1995 година. В момента той работи паралелно с IPv4 в гръбнака на Интернет и във всички доставчици от ниво 1 – свързаните директно към гръбнака. В някои страни, например в Холандия и в по-голямата част от Азия крайните клиенти вече получават IPv6 свързаност. Повечето големи сайтове имат едновременно IPv4 и IPv6 адреси. Според статистиката на Google обаче в края на 2013 година само 2% от заявките към него са направени по новия протокол, което показва известно забавяне при неговото разпространение.

Като основно предимство на IPv6 се счита огромното адресно пространство, което се дискутира по-подробно в следващата точка. От него обаче се очаква да реши и други проблеми – има повече функции за автоматично назначаване на адреси, механизми за качество на обслужването, сигурност, мобилност и други.

### 6.1 Структура на IPv6 адрес.

IPv6 адресите са 128 битови, за разлика от 32 битовите IPv4 адреси. Броят достъпни IP адреси при новият протокол е $2^{128} = 3.4\times10^{38}$ адреса. Това е приблизително $665\times10^{21}$ адреса на квадратен метър от площта на земята. Приетият запис на IPv6 адресите е с шестнадесетични цифри, групирани по четворки и разделени с двоеточие, например:

1234:5678:9ABCD:EF01:2345:6789:ABCD:EF01

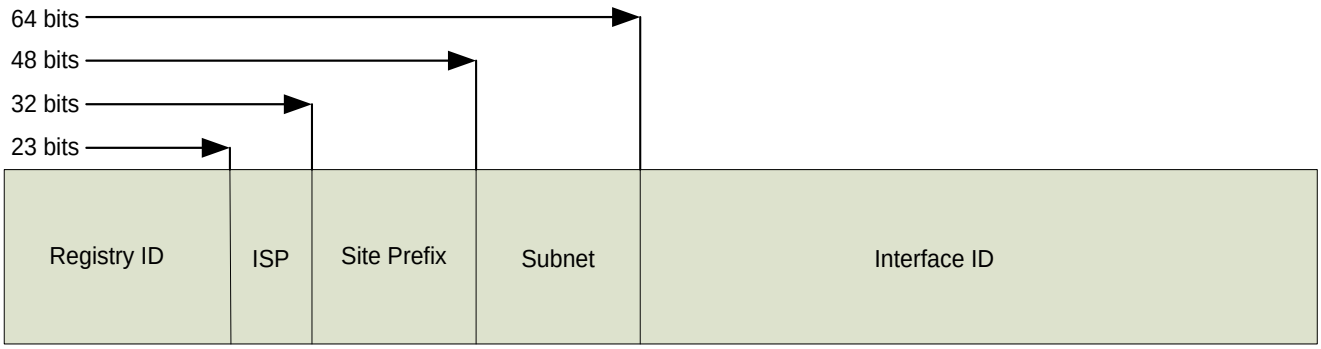
Всяка шестнадесетична цифра описва 4 бита със стойност от 0 до F. 128 бита се описват с 32 шестнадесетични цифри, разделени на осем четворки.

Понеже адресите са твърде много, голяма част от цифрите в адресите в началото се очаква да бъдат нули. Затова са предвидени следните правила за съкратено записване:

- Водещите нули могат да се пропускат, т.е адрес:
  1234:0567:0089:0001:0234:0078:0001:0012 може да се запише като:
  1234:567:89:1:234:78:1:12.
- Една от няколко поредици от четворки с нули, може да се замести с “::”, напр:
  1234:0000:0000:5678:0000:9ABC:DEF1:2345 може да се запише като:
  1234::5678:0:9ABC:DEF1:2345 или 1234:0:0:5678::9ABC:DEF1:2345
- Не може да заместим и двете нулеви поредици с ::, защото няма да знаем къде (в примера) са четири и къде – осем нули.
- Последните 32 бита за по-добро разбиране и съвместимост могат да се изписват като IPv4 адреси, например 876A:130B = 135.106.19.11.

Вместо мрежова маска при IPv6 се използва т. нар. префикс, напр: 2345:BA23:0007::/50, което означава „първите 50 бита от адреса са:...“. Всъщност показаните 12 шестнадесетични цифри описват 48 бита. Префиксът /50 означава, че следващите 2 бита са 0. Ако префиксът беше /64, това би означавало, че следващите (след 48 битовия префикс) 16 бита са 0.

Текущият начин на раздаване на IPv6 адреси е показан на фиг. 6.1.




**Фиг. 6.1. Разпределение на IPv6 адрес.**

- 48 бита Routing Prefix, състоящ се от:
  - 23 бита Registry ID – номер на регистратора;
  - 9 бита ISP Prefix – номер на Интернет доставчика.
- 16 бита Site Prefix – номер на клиента.
- 16 бита Subnet Prefix – номер на подмрежата.
- 64 бита Interface ID – номер на компютъра.

Забележете, че всяка мрежа или подмрежа при тази схема, дори всеки домашен клиент ще има $2^{64}$ адреси, при положение, че сегашния Интернет има общо $2^{32}$ адреса, тоест при тази схема всеки дом или офис ще бъде много пъти по-голям като адресно пространство от целия сегашен Интернет. В момента организациите получават /48 мрежи, което позволява 65535 подмрежи с по приблизително $1,8X10^{19}$ компютри във всяка подмрежа, а частните клиенти получават една /64 мрежа. Възможно и много вероятно е този начин на раздаване на адреси да се промени в близко бъдеще, поради прекаленото разхищение на адреси.

Адресното пространство, заделено за глобално достижими от Интернет адреси са адресите, започващи с най-леви три бита 001 или това са адреси, започващи с най-лява четворка 2000:: до 3FFF:: - тези адреси в терминологията се наричат Global Unicast. От тях също има такива, които са определени за специфични цели, но те са извън обхвата на тази книга.

### 6.2 Специални IPv6 адреси

Адрес 0000:0000:0000:0000:0000:0000:0000:0000/128 или съкратен запис ::/128 се нарича Unspecified или неопределен адрес. Той може да се използва като източник на пакет, когато устройството все още няма назначен адрес.

Адрес 0000:0000:0000:0000:0000:0000:0000:0000/0 или съкратен запис ::/0 се нарича Default Route или път по подразбиране и има същия смисъл като IPv4 адреса 0.0.0.0.

Адрес 0000:0000:0000:0000:0000:0000:0000:0001 или съкратен запис ::1 се нарича Loopback – по подобие на адреса 127.0.0.1 при IPv4 изпратен пакет до този адрес стига до собствения компютър, независимо какъв или какви други IPv6 адреси има.

Адреси започващи с FC00:: - FDFF:: се наричат Unique Local Addresses и имат смисъла на IPv4 частните адреси.

Адреси започващи с FE80:: - FEDF:: се наричат Link Local и пакети изпратени до такъв адрес никога не преминават през маршрутизатор. Обикновено всеки компютър или устройство освен глобалния си IPv6 адрес, с който е видим в Интернет има и автоматично назначен Link local адрес на всеки свой мрежов интерфейс, който се използва за локална комуникация.

Адреси, започващи с FEC0:: – FEFF:: се наричат Site Local и не се препращат извън мрежата на организацията. Те служат за изграждане на услуги, които са достъпни само за дадената организация или Интернет доставчик.

Адреси започващи с FF00:: - FF1E:: са заделени за многоцелеви (Multicast) предавания, подобно на IPv4 адресите от клас D.

Адреси 0000:0000:0000:0000:0000:FFFF/96 или съкратен запис ::FFFF/96 се наричат IPv4 mapped и служат за по-лесен преход от IPv4 към IPv6 адресиране, например IPv4 адресът 192.168.1.1 може да се мигрира в IPv6 и да се представи като ::FFFF.192.168.1.1.

При IPv6 няма Broadcast предаване на данни. Към традиционните предавания Unicast и Multicast е добавен и нов тип – Anycast, който означава предаване на данни до най-близкия от дадена група получатели. Anycast адресите са от глобалното адресно пространство, но след префикса завършват с нулеви битове. Те са използваеми за разпределяне на услугите, натоварването и отказоустойчивостта между група сървъри с еднаква конфигурация, разположени на различни места по света или в рамките на организацията.

Пример за Anycast адрес е: 2001:1234:5678:9ABC:0000:0000:0000:0000/64.

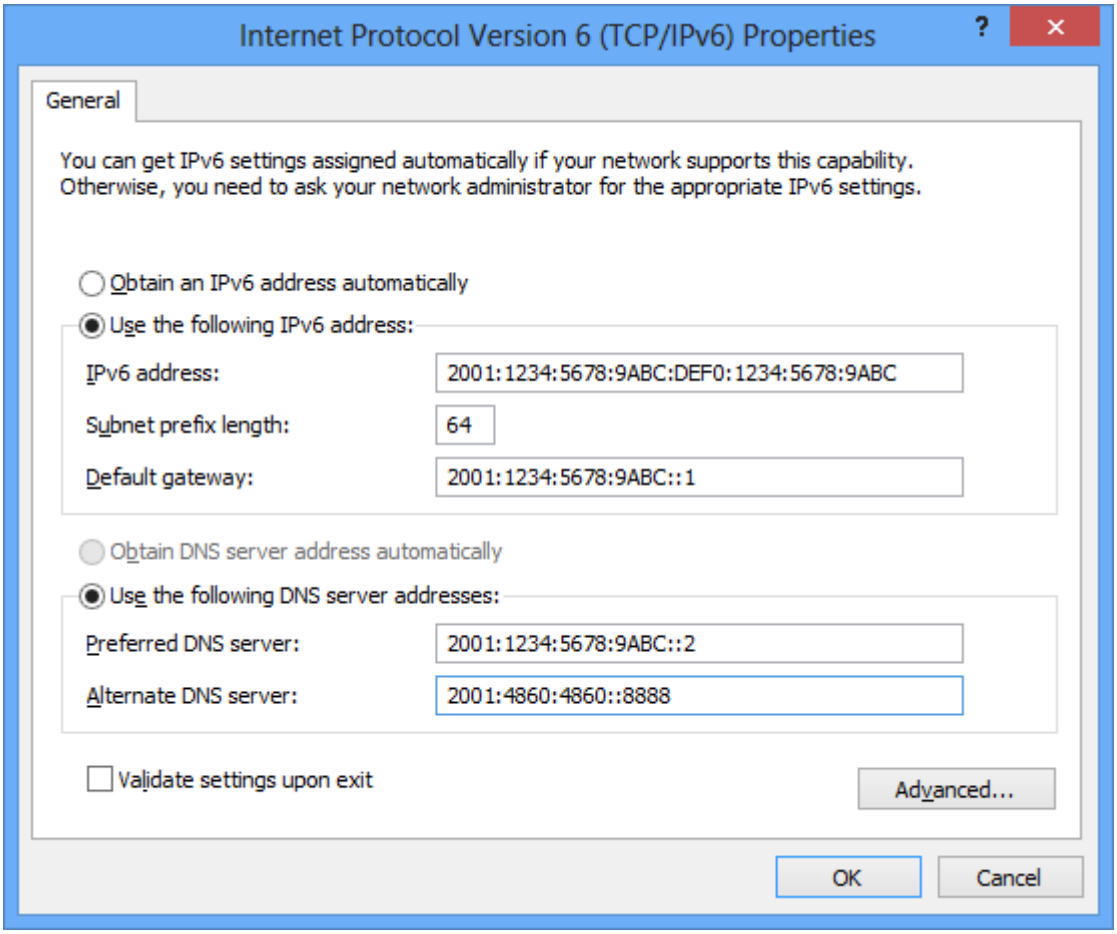
## 6.3 Методи за назначаване на IPv6 адреси

При IPv4 методите за назначаване на адрес са статично – когато ръчно напишем адреса в конфигурацията на устройството и динамично, когато разчитаме на сървър, работещ с протокола DHCP, който да назначи IP адрес и свързаните с него параметри.

Тези два типа са достъпни и при IPv6, но към тях са добавени и допълнителни начини, които да автоматизират назначаването на адреси без наличието на специален сървър или без необходимостта от ръчно писане на всички цифри от адреса.

### 6.3.1 Статично назначаване на IPv6 адрес

На фигура 6.2 е показан екранът за статично назначаване на IPv6 адреси във Windows 8.




**Фиг. 6.2. Статично назначаване на IPv6 адрес и параметри.**

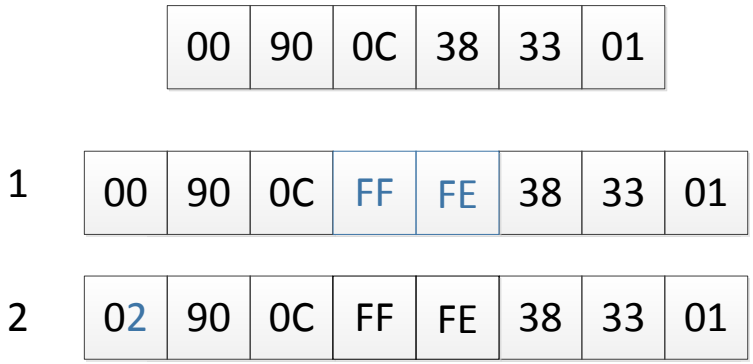
Параметърът Default Gateway (шлюз по подразбиране) има същото значение, като при IPv4 – всички пакети, които не са за нашата мрежа се предават на този възел. Параметърът Prefix Length указва колко от левите битове на IPv6 адреса са номер на мрежа, а DNS сървърите, обяснени по-подробно в главата за приложно ниво

отговарят за превеждането от имена на сайтове (например www.tugab.bg) в адреси. Адресите, показани на фигурата са примерни.

### 6.3.2 Назначаване само на номер на мрежа – EUI-64 Interface ID

Идеята на този метод е да се назначи статично само първите 64 бита на адреса, означаващи номера на мрежата, а компютърът да изчисли автоматично десните си 64 бита. Не всички операционни системи позволяват подобна конфигурация, но идеята е използвана в следващия метод за автоматично конфигуриране на адреси.

Начинът на изчисляване на адреса е показан на фигура 6.3.




**Фиг. 6.3. Механизъм EUI-64**

Втората част от адреса се изчислява на базата на MAC адреса на компютъра, който би следвало да е уникален. В примера MAC адресът е 0090.0C38.3301. На първата стъпка дванадесетте шестнадесетични цифри на MAC адреса се разделят на две порции по шест цифри и между тях се вмъкват цифрите FFFE. Това е резервирана стойност, коят производителите не могат да вграждат в истинските си устройства и по този начин може да се установи, че адресът е генериран автоматично. На втората стъпка се инвертира стойността на седмия бит отляво надясно – битът универсален/локален адрес (universal/local, U/L). На истинските устройства този бит има стойност 0.

Действието на механизма при операционна система Cisco IOS е показано на следващите фрагменти.

```
Router#show interface fastEthernet 0/0
FastEthernet0/0 is administratively down, line protocol is down
  Hardware is Lance, address is 0090.0c38.3301 (bia 0090.0c38.3301)…
```

Командата показва MAC адреса на интерфейса, който в случая е 0090.0c38.3301.

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fastEthernet 0/0
Router(config-if)#ipv6 address 2001:1234::/64 eui-64
```

С тези команди се конфигурира първата част от IPv6 адреса да бъде 2001:1234:0000:0000, а втората част да се генерира по алгоритъма EUI-64.

```
Router#show ipv6 interface
FastEthernet0/0 is administratively down, line protocol is down
  IPv6 is enabled, link-local address is FE80::290:CFF:FE38:3301 [TEN]
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:1234::290:CFF:FE38:3301, subnet is 2001:1234::/64 [EUI/TEN]…
```

Командата показва IPv6 адресите на интерфейса. Виждаме два такива – автоматично генерираният локален (Link-Local) адрес FE80::290:CFF:FE38:3301 и конфигурираният глобален адрес 2001:1234::290:CFF:FE38:3301. Вижда се, че и на двата десните 64 бита съвпадат и имат стойност 0290:CFF:FE38:3301, получена от MAC адреса по алгоритъма EUI-64.

### 6.3.3 Механизъм за автоматично определяне на целия адрес (Stateless Autoconfiguration)

При този механизъм компютърът изчислява автоматично по алгоритъма EUI-64 дясната част на своя адрес, след което изпраща запитване към маршрутизатора, който му връща своя префикс. Компютърът назначава получения префикс за лява част на адреса си и така получава пълния си адрес автоматично, без необходимост от DHCP сървър.

### 6.3.4 Назначаване на адрес чрез DHCP сървър

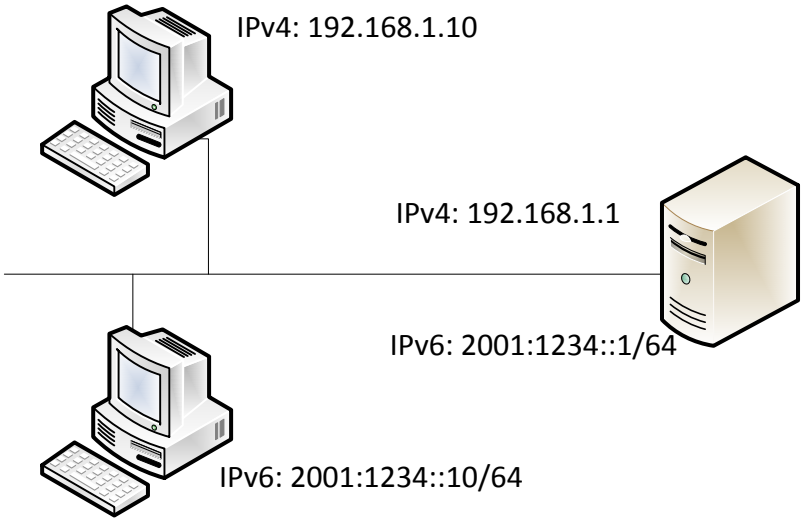
При този метод, както и при IPv4 е необходимо наличието на специален сървър за назначаване на адреси, работещ според протокола DHCP (Dynamic Host Configuration Protocol), обяснен по-подробно в главата за приложно ниво. На този сървър се задават правила, при включването компютърът търси сървър, който отговаря, връщайки необходимите адресни параметри. Методът се нарича още Stateful Autoconfiguration.

## 6.4 Механизми за съвместимост

Невъзможно е да се помисли, че новият протокол може едновременно да бъде приложен в цялата мрежа Интернет. Някои устройства никога няма да могат да бъдат мигрирани към IPv6 поради ограничения в техния софтуер и/или хардуер. Предполага се че едновременното функциониране на двата протокола IPv4 и IPv6 ще продължи десетки години. За това са предвидени различни механизми за съвместимост, които да позволят едновременната работа на устройства по двата протокола в мрежата. Те могат да се използват самостоятелно или в комбинация в зависимост от поставените цели.

### 6.4.1 Двоен стек (Dual Stacking)

Това е един от основните механизми, който вече се използва от големите доставчици на услуги в Интернет и ще бъде използван дълго време, докато не се направи пълна миграция. Механизмът е представен на фигура 6.4.

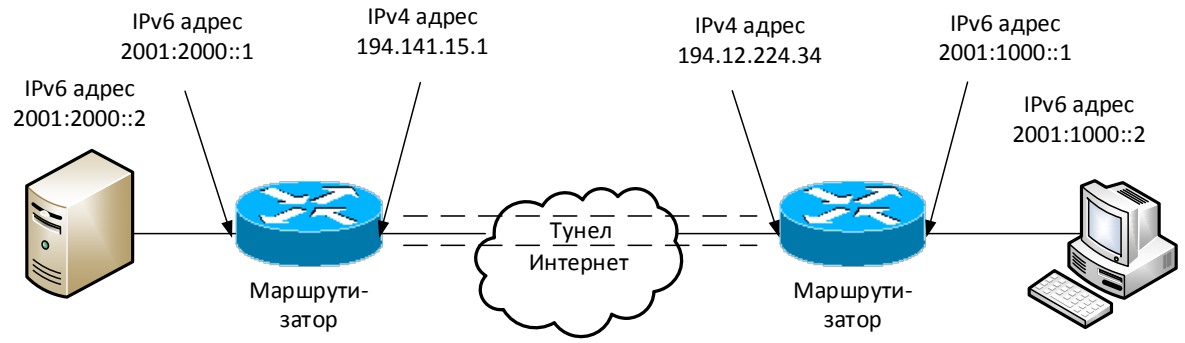



**Фиг. 6.4. Механизъм двоен стек.**

Идеята е на устройствата, даващи услуги на мрежата да се инсталират и двата протокола, като по този начин и вече мигрираните компютри, и тези които още не са мигрирали ще могат да изпращат заявки към сървъра и да използват услугите му. Това показва, че механизмът е подходящ и в локални мрежи по време на процеса на мигриране, но доста компании в Интернет, като Google, Facebook и други отдавна са приложили механизма на своите сървъри, като по този начин ги правят достъпни за клиенти, работещи и с двата протокола. Разбира се при този механизъм устройствата, работещи с различни протоколи в една мрежа няма да могат да си изпращат информация директно.

### 6.4.2 Тунелиране (Tunneling)

Тунелирането е един отдавна известен механизъм. При миграцията към IP версия 6 той може да се използва например когато всички клонове на дадена компания в различни градове или страни са свързани през Интернет и компанията реши да мигрира към новия протокол. Ако обаче Интернет доставчиците на някои от клоновете все още не предоставят свързаност чрез IPv6, то този клон все пак може да мигрира, използвайки механизма на тунелиране – да създаде IPv6 тунел между своя маршрутизатор и маршрутизатора на централния офис през IPv4 мрежата на доставчика. Това е показано на фигура 6.5.




| IPv4 Заглавна част | IPv6 Заглавна част | |
|---|---|---|
| От 194.12.224.34<br>До 194.141.15.1 | От 2001:1000::2<br>До 2001:2000::2 | Пакет |

**Фиг. 4.5. Тунелиране.**

Когато компютърът от отдалечения офис вдясно на фигурата предаде пакет към сървъра в централния офис, той изработва IPv6 заглавната част и изпраща пакета в локалната мрежа към своя маршрутизатор. За да изпрати пакета през тунела, маршрутизаторът изработва нова заглавна част на IPv4 протокол, за да може пакетът да премине през IPv4 мрежата и в нея поставя IPv4 адресите, определящи началото и края на тунела. Така пакетът достига до другия маршрутизатор, който премахва допълнителната заглавна част и полученият IPv6 пакет се предава до сървъра. Пкатетите в обратна посока се предават аналогично.

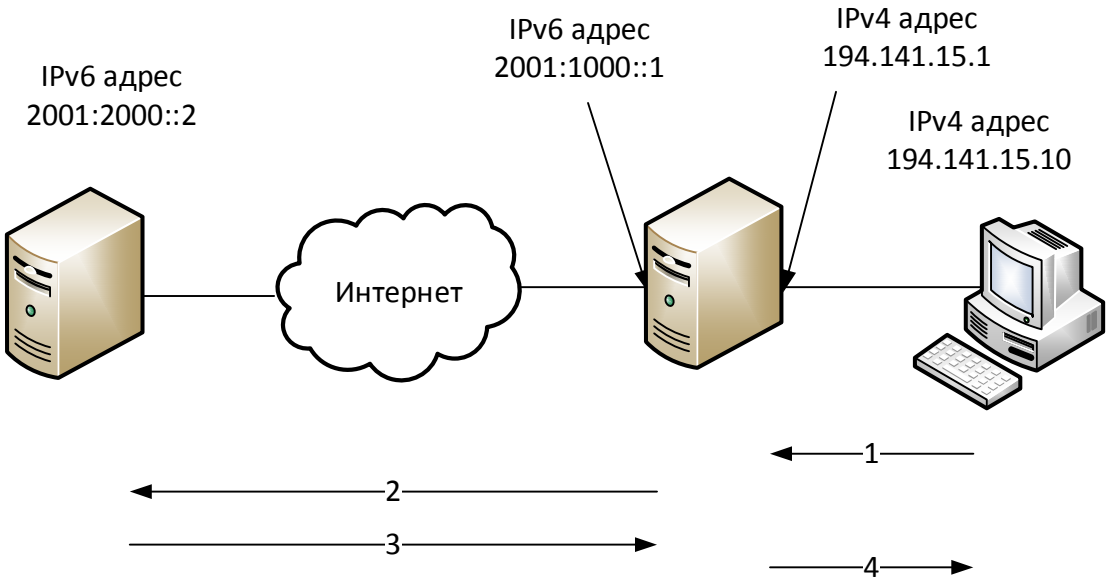
Съществуват няколко вида тунели, предназначени за целите на съвместимост на двата IP протокола. Част от тях са:

- Статичен тунел (IPv6-over-IPv4) – при него параметрите на тунела, IPv4 и IPv6 адресите се конфигурират ръчно и не подлежат на автоматична промяна. Предполага двуточкова конфигурация.
- Динамично тунелиране (Dynamic 6to4 tunneling) – разработен за топология звезда – централен офис, свързващ динамично много отдалечени офиси, чиито адреси могат да се променят. Използва резервираното адресно пространство 2002::/48, при което останалите битове от IPv6 адреса се образуват автоматично от съответния IPv4 адрес.
- ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) тунели – предназначени за предаване на IPv6 пакети между устройства с двоен стек през IPv4 мрежи.
- Teredo тунели – предназначен за предаване на данни между IPv6 устройства, свързани към IPv4 мрежа. Използва се специалния префикс 2001:0::/32.
- Tunnel brokers – Външни (в Интернет) сървъри, позволяващи динамично установяване на тунел между няколко страни, използващи протокол TSP (Tunnel Setup Protocol).

### 6.4.3 Превод на протоколи (Protocol Translation)

Услугата превод на протоколи е предназначена за осъществяване на двупосочна комуникация между IPv4 и IPv6 възел. Той се реализира по два начина:

- Чрез прокси сървър – посредник, който комуникира директно с двете страни, както е показано на фигура 4.6.




**Фиг. 4.6. Прокси сървър**

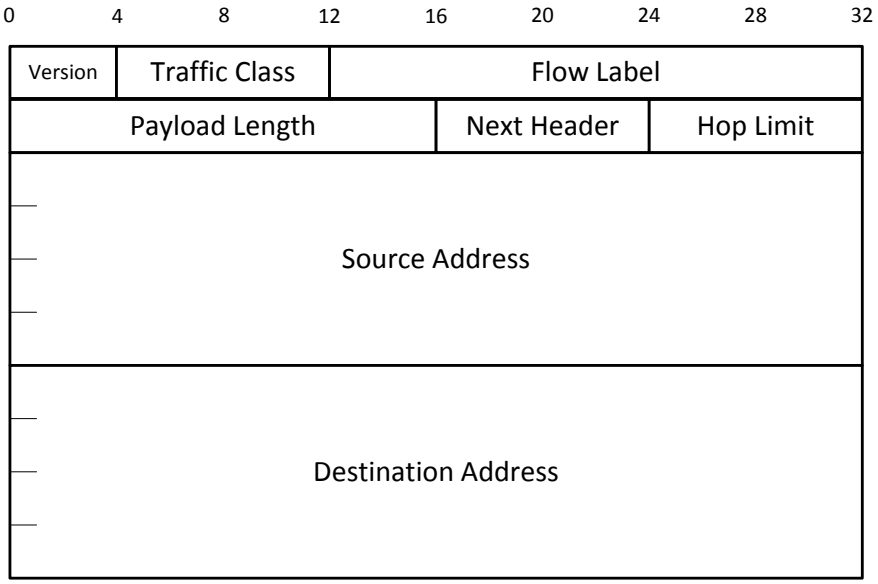
При предването през прокси сървър мрежата на клиента е IPv4, а Интернет – IPv6. Клиентът предава заявката за даден Интернет ресурс, например web страница до прокси сървъра чрез IPv4. Прокси сървърът изпраща заявка чрез IPv6 към web сървъра, прочита страницата и я предава до клиента чрез IPv4. Някои Интернет услуги, като web, файлов трансфер и електронна поща по дизайн поддържат прокси, други – например видео в реално време не са.

- Чрез подмяна на заглавните части на протоколите – най популярния в момента механизъм се нарича NAT64. При него посредникът (най-често маршрутизатора) не следи потока данни между клиента, а просто подменя заглавните части на IPv4 пакетите с IPv6 такива и обратно за входящите пакети. Обменът на пакети като тип и поредица е същия, като в горния пример.

Разликата между двата механизма е в това, че при сценария с прокси сървър цялата информация (файл, уеб-страница) се сглобява в посредника и в последствие се препредава към клиента, докато при подмяната на заглавните части на пакетите посредникът предава информацията пакет по пакет, а целият ресурс се сглобява при клиента. Моделът с прокси сървър предоставя по-голяма степен на сигурност и на контрол върху потребителите.

### 6.5 Сравнение на заглавните части

Създателите на IPv6 с гордост заявяват, че въпреки че IP адресите в новия протокол са четири пъти по-дълги, заглавната част на протокола е само два пъти по-дълга. При IPv6 тя е 40 байта и не може да бъде различна, за разлика от IPv4, където може да има опции. Това че заглавната част е еднаква за всички пакети означава, че обработката на пакетите в маршрутизаторите е по-опростена и се използват по-ефективно ресурсите им. Структурата на заглавната част на IPv6 е показана на фигура 6.7.




**Фиг. 6.7. Заглавна част на IPv6.**

Полетата са както следва:

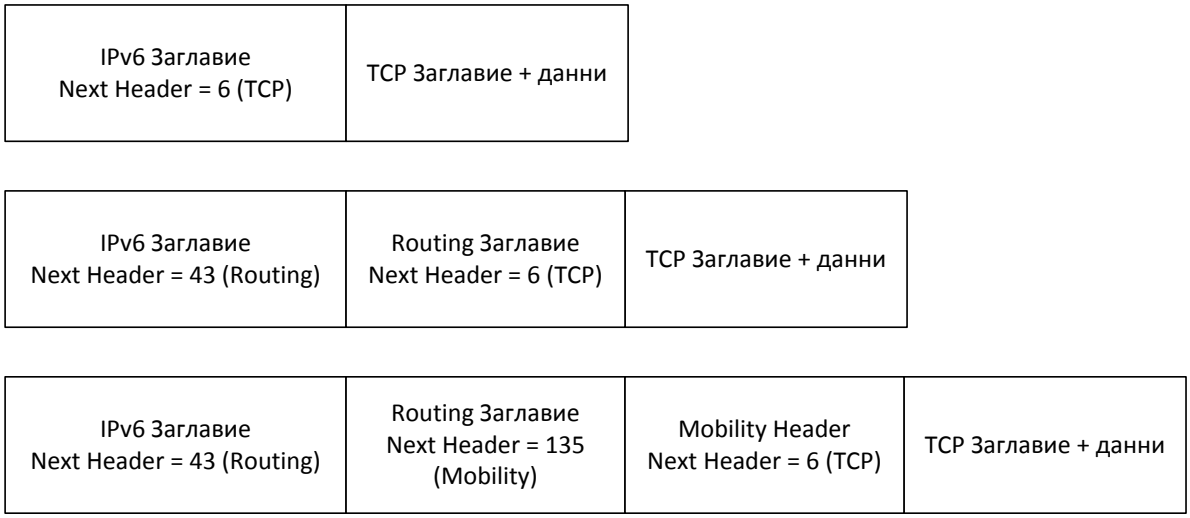
- Версия (Version) – четири-битово поле, показващо версията на протокола – в случая има стойност 6.
- Клас на трафика (Traffic Class) – осем-битово поле. Замества полето тип на услугата (Type of Service) и служи за дефиниране на приоритети.
- Етикет на потока (Flow Label) – 20 битово поле, служещо да увеличи възможностите за дефиниране и обработка на качество на обслужването.
- Дължина на данните (Payload Length) – 16 битово поле, показващо броя байтове на съдържанието на пакета без заглавната част.
- Следващо заглавие (Next Header) – осем-битово поле, съдържащо указател към следваща заглавна част, ако има такава. Концепцията на допълнителните заглавни части е обяснена в следващата точка.
- Ограничение на преходите (Hop Limit) – осем-битово поле, указващо максималния брой преходи, които може да направи пакета, преди да бъде изхвърлен. Намалява се във всеки възел с 1, подобно на Time-to-Live при IPv4.

- Адрес на източника (Source Address) – 128 битово поле, съдържащо IPv6 адреса на източника.
- Адрес на получателя (Destination Address) – 128 битово поле, съдържащо адреса на получателя.

Разликите при двата протокола са: при IPv6 мрежата не осигурява фрагментация, затова са премахнати полетата идентификация, флагове и отместване на фрагмента. Премахната е контролната сума на заглавната част, което означава че маршрутизаторът няма задача да я изчислява във всеки възел.

### 6.6 Заглавия за разширения (Extension Headers)

За да се осигури еднаква дължина на заглавната част при IPv6 са премахнати опциите, а допълнителни функции могат да се осъществяват с вмъкване на допълнителни заглавни части след основната. Броят и дължината на тези заглавни части не са строго определени, което позволява вграждането на няколко допълнителни функции в пакета. Идеята е, че в полето следваща заглавна част (Next Header) се записва число, указващо типа на следващото заглавие или числото 59, ако липсва такова. Подобно на полето Protocol при IPv4 има запазени числа за различните протоколи, например 6 означава TCP, а 17 – UDP. На фигура 6.8 са показани примери за пакети с по една или няколко допълнителни заглавни части.

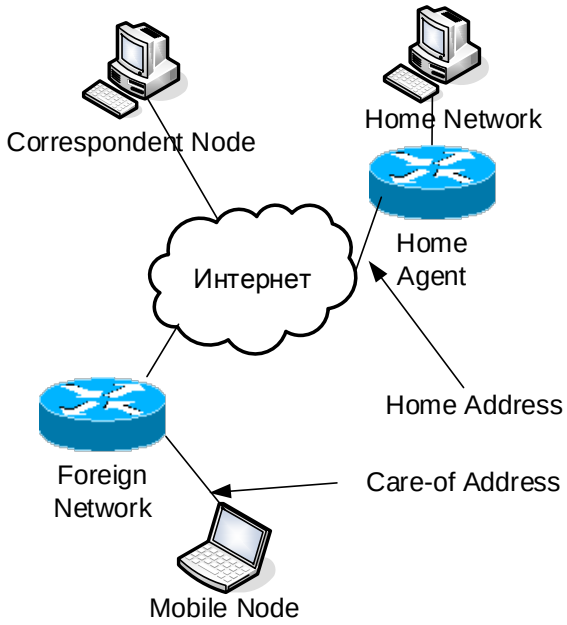



**Фиг. 6.8. Допълнителни заглавни части.**

### 6.7 Мобилност при IPv6

Една от допълнителните функции на новия протокол е функцията мобилност - възможност, позволяваща на даден компютър да се идентифицира с един и същ адрес в Интернет независимо от неговото конкретно местоположение. Чрез тази функция може да се осигури възможност за връзка независимо от движението на потребителя,

свързването му към различни точки на мрежата и това става автоматично, без нужда от потребителска намеса. Основните компоненти на механизма са показани на фиг. 6.9.




**Фиг. 6.9. Основни компоненти на функцията мобилност**

Всеки възел, който се нуждае от функцията мобилност трябва да има глобален уникален IPv6 адрес (Home Address, HA). Този адрес се използва за комуникация с мобилното устройство. Когато е свързано в домашната си мрежа, то получава този адрес и комуникацията от външния свят с устройството е директна. Когато устройството се намира в друга мрежа, пакетите изпращани до този адрес се прихващат от домашния агент (Home Agent, HA) и се препращат към текущото местоположение на мобилното устройство.

Домашният агент е маршрутизатор или компютър, който осигурява регистрацията на мобилните устройства, намиращи се извън домашната мрежа и техните текущи IPv6 адреси.

Care-of адрес е адресът, назначен на мобилното устройство, когато то е във външна мрежа. Този адрес се регистрира в домашния агент и се използва за препращане на трафика, изпратен до домашния адрес на мобилното устройство. Този адрес може да се променя при движението на мобилното устройство от мрежа в мрежа.

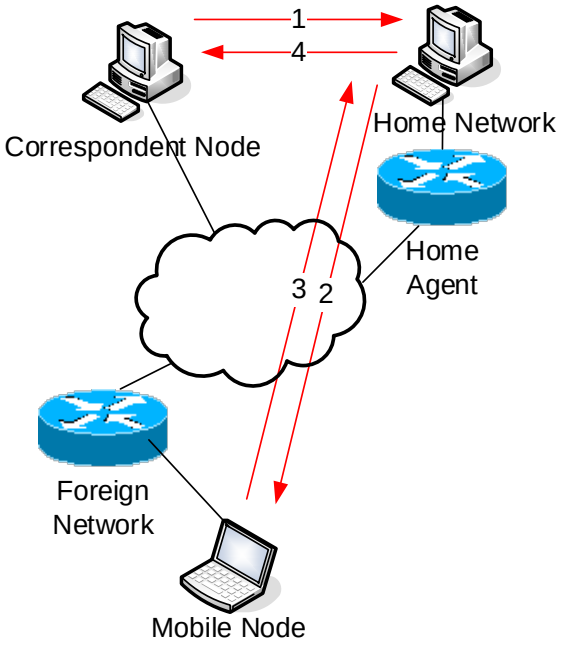
Кореспондент (Correspondent Node, CN) е произволен компютър, свързан към Интернет, който желае да обменя информация с мобилното устройство.

Използвайки такава структура винаги когато мобилният компютър е извън домашната си мрежа и получи нов адрес според текущото си местоположение, той регистрира този адрес в домашния си агент. Всеки компютър в Интернет, желаещ да комуникира с мобилното устройство изпраща заявка до домашния агент.

Възможни са два механизма на комуникация:

- Двупосочно тунелиране – при него домашният агент получава пакетите за мобилното устройство, препраща ги през специално изграден тунел към текущия адрес на устройството, през тунела получава обратния трафик и го пренасочва обратно към кореспондента.

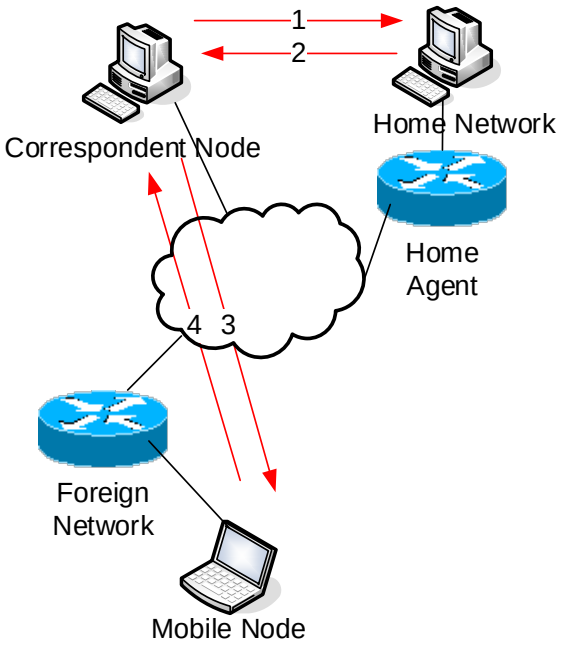
Този процес е показан на фигура 6.10.




**Фиг. 6.10. Двупосочно тунелиране**

- Директна маршрутизация – домашният агент получава заявката от кореспондента и му връща съобщение за пренасочване на трафика, съдържащо текущия адрес на мобилното устройство, който се използва за следващите пакети. Процесът е показан на фигура 6.11.

Първият подход има две предимства – той не изисква специални възможности за пренасочване на трафика в кореспондента и скрива мобилността, т.е. кореспонентът не разбира, че мобилният възел е на друго място, нито къде е. Недостатъците са увеличаването на натоварването на мрежата и натоварването върху домашния агент, което може да изисква значителни ресурси при много мобилни възли.




**Фиг. 6.11. Директна маршрутизация.**

# 7. Маршрутизация

Маршрутизацията е функцията на мрежовите устройства чрез която те определят най-добрия път за пакетите. Терминът „най-добър път“ може да означава различни неща, в зависимост от избраната оценка на трасетата – например най-къс, най-бърз или най-евтин път. В общия случай за да стигне пакетът от източника до получателя в Интернет той трябва да премине през няколко различни маршрутизатора. Всеки от тях изпълнява като минимум следните действия:

1. Получава пакета на входящия си интерфейс.
2. Проверява дали MAC адресът съвпада с неговия.
3. Проверява за грешки, изчислявайки контролна информация и сравнявайки стойността с приетата. При наличие на грешка пакетът се изхвърля, иначе:
4. Премахва заглавната и завършващата част на каналния протокол.
5. Извлича IP адреса на получателя.
6. Сравнява го със знанията в таблицата си, като на всеки ред умножава адреса по мрежовата маска, за да отдели номера на мрежата и определя дали има съвпадение. Структурата на таблицата е показана в следващите точки.
7. При липса на съвпадение с някой от редовете пакетът се изхвърля. При наличие на съвпадение на даден ред се определя къде ще се изпрати пакета.
8. Намалява се TTL с 1, изчислява се нова контролна сума (при IPv4) и се записва в IP заглавната част.
9. Намира се MAC адреса на следващия получател, създава се нова заглавна част на канално ниво.
10. Изчислява се нова стойност за проверка за грешки и се записва отзад на кадъра.
11. Пакетът се поставя в опашка, където изчаква реда си за предаване.

Всеки път когато пакет преминава през маршрутизатор, MAC адресите в заглавната част на канално ниво се променят и те указват текущия източник и получател, а IP адресите в повечето случаи (освен ако не се изпълнява техниката NAT) не се променят и те указват началния източник и крайния получател, т.е. MAC адресите имат локално значение и играят роля в конкретната мрежа, а IP адресите имат глобално значение и идентифицират устройствата в цялата мрежа Интернет.

В Интернет всеки маршрутизатор независимо от останалите изпълнява горния алгоритъм и взима решение къде да изпрати пакета. Възможно е при неправилни знания два маршрутизатора да взимат различни решения и да си предават даден пакет един на друг докато изтече времето му на живот. Съществуват други видове мрежи, например MPLS (Multi Protocol Label Switching), при които решението за целия път на пакета през мрежата се взима еднократно от граничния маршрутизатор, а останалите просто предават пакета по вече избрания път.

### 7.1 Маршрутизираща таблица.

Както беше споменато, за да определи пътя на пакетите, маршрутизаторът трябва да запазва знания за мрежите. Тези знания се запазват в маршрутизираща таблица (Routing table). Примерна структура на такава таблица е показана на фигура 7.1.

| Научен от | Адрес на мрежа | Метрика | Изпращане |
|---|---|---|---|
| C | 192.168.1.0/24 | 0 | Ethernet 0 |
| C | 192.168.5.0/24 | 0 | Ethernet 1 |
| C | 172.16.0.0/16 | 0 | Serial 0 |
| S | 10.0.0.0/8 | 1 | 192.168.1.2 |
| R | 192.168.8.0/24 | 2 | 10.0.0.2 |

**Фиг. 7.1. Примерна маршрутизираща таблица.**

В колоната „Научен от“ се отбелязва начинът, по който маршрутизаторът е научил информацията за тази мрежа. Повече детайли за начините на научаване на информацията за различните мрежи са показани в следващите точки. В колоната „Адрес на мрежа“ се записва адресът на мрежата, заедно с мрежовата маска. В колоната „Метрика“ се записва число, определящо колко е добър този път, според критериите, приложени към дадения маршрутизатор. В колоната „Изпращане“ се записва къде трябва да се изпрати този пакет, за да стигне до указаната мрежа. Забележете, че в тази колона може да стои IP адрес на следващия маршрутизатор по пътя или интерфейс на локалния маршрутизатор, където трябва да се изпрати пакета. В реалните маршрутизиращи таблици могат да се съдържат и още параметри, но те са извън нашия обхват.

Когато се определя къде ще се изпрати даден пакет, неговият IP адрес на получател се сравнява подред с данните на всеки ред за тази колона, като първо адресът се умножава по мрежовата маска, за да се отдели номера на мрежата и остатъкът се проверява дали съвпада със съдържанието на реда. Ако на някой ред се получи съвпадение, то се проверява последната колона и пакетът се изпраща нататък. Ако се достигне последния ред и не бъде открито съвпадение, пакетът се изхвърля.

В таблиците на повечето маршрутизатори като последен ред[2] стои специалният IP адрес „път по подразбиране“ (Default Route), който се обозначава като IP адрес 0.0.0.0 с маска 0.0.0.0. За маршрутизатора това означава всички пакети, които не съвпадат с

[2] При някои операционни системи не точно редът на записите е определящ, а така нареченото правило “Най-дълго съвпадение“, т.е. ако пакетът съвпада с два или повече реда се изпраща според реда, за който съвпадат най-много битове.

никой от горните редове. Това става следвайки логиката на умножение и сравнение – какъвто и да е IP адресът, умножавайки го по мрежова маска 0.0.0.0 резултатът е 0.0.0.0, което се сравнява с номера на мрежата – така всеки пакет има съвпадение с този ред.

### 7.2 Статична и динамична маршрутизация

В зависимост от начина на попълване на знанията в таблицата различаваме статична и динамична маршрутизация. При статичната редовете в таблицата се попълват ръчно от администратор. При динамичната отделните маршрутизатори обменят знания помежду си, чрез избран маршрутизиращ протокол и така автоматично попълват таблиците си. Възможно е комбиниране – някои редове да се зададат ръчно, други да се научават автоматично. Обикновено статичните редове имат приоритет пред динамичните, но е възможно да се конфигурира и обратното.

Статичната маршрутизация има няколко предимства пред динамичната. Тя е по-икономична, защото при нея не се налага устройствата да си предават служебна информация и знания за мрежите. Тя е и по-сигурна, защото при динамичната е възможно при неправилна конфигурация даден маршрутизатор случайно или нарочно да обявява неверни знания и останалите да ги приемат като истина.

Динамичната маршрутизация има едно важно предимство – тя може автоматично да адаптира мрежата към промяната на топологията, което при статична маршрутизация налага администраторска намеса. При големи и сложни мрежи е почти невъзможно конфигурирането със статична маршрутизация.

Голяма част от домашните и малките фирмени мрежи имат топологията, показана на фигура 7.2.


**Фиг. 7.2. Мрежа с една връзка към Интернет**

При тези прости мрежи с една единствена връзка към Интернет (на английски се наричат Stub network) се предпочита статична маршрутизация. При тях обикновено таблицата се състои само от два реда – път към собствената мрежа, сочещ към вътрешния интерфейс и път по подразбиране (Default Route), сочещ към външния интерфейс.

### 7.3 Маршрутизиращи протоколи

Следващите точки са посветени на динамичната маршрутизация, която изисква стартирането на динамичен маршрутизиращ протокол на устройствата, които си обменят знания. Маршрутизиращият (routing) протокол определя как устройството изчислява най-добрия път, какви критерии използва за оценката, каква информация предава, до кои други устройства я предава, кога я предава и доста други параметри. Разбира се в повечето случаи всички маршрутизатори от дадена мрежа трябва да работят с един и същ протокол, за да могат да обработват еднозначно информацията. В някои специални случаи е възможно смесването на протоколи, но това е извън обхвата на настоящата тема.

Освен маршрутизиращите протоколи, които се явяват служебни за мрежата, защото помагат мрежата да върши своята работа, в нея работят и други протоколи – тези, които пренасят данните на потребителите. Те се наричат „маршрутизирани“ (Routed) протоколи и са полезните от гледна точна на потребителя, защото мрежата е създадена за да пренася потребителските данни. Примери за такива протоколи са IPv4 и IPv6.

Както беше споменато оценката на даден път се прави, като за всеки възможен път се изчислява метрика. Метриката е число, определящо доколко е добър даден път. Обикновено колкото по-малко е числото, толкова по-добър е пътят. Критериите за оценка на маршрутите, използвани в реалните маршрутизиращи протоколи включват: скорост на трасето, времезакъснение на пакетите, надеждност на предаването, натоварване на интерфейсите, брой на преходите и други. Някои протоколи използват само един критерий за оценка, други могат да изработват комплексна метрика, изчислявана от няколко критерия едновременно.

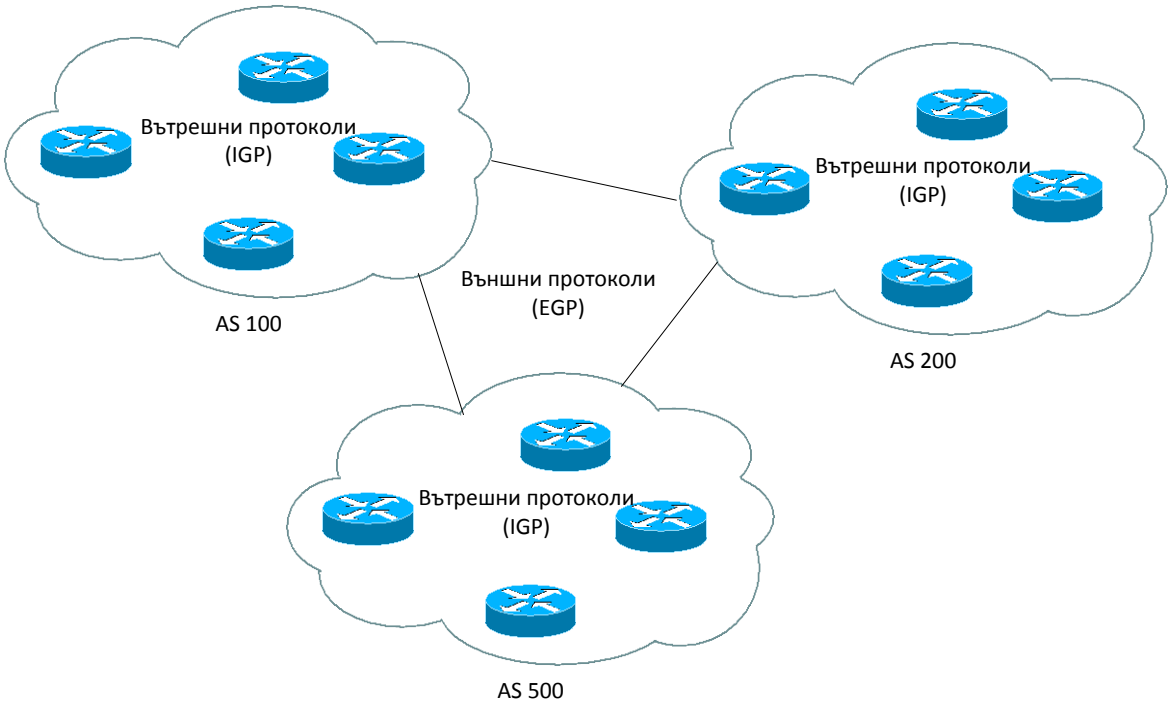
Независимо от използвания протокол, в началото маршрутизаторът има информация само за директно свързаните към него мрежи. За тях той научава от IP адресите и маските, които са назначени на неговите интерфейси, като умножи IP адреса си по мрежовата маска и получената мрежа записва в таблицата си като директно свързана. Такава мрежа се означава с метрика 0 – маршрутизаторът има най-добрия път до нея и никой друг път не може да е по-добър. За тези мрежи той предава пакетите на съответния си интерфейс.

#### 7.3.1 Външни и вътрешни протоколи

Не е възможно, а и не е необходимо всички маршрутизатори в Интернет да предават информация до всички останали. Интернет е разделен на отделни участъци, наречени „Автономни системи“ (Autonomous Systems, AS). Автономната система се дефинира като група маршрутизатори под обща администрация, споделящи общи протоколи и политики. Обикновено всички маршрутизатори на един Интернет доставчик, заедно с тези на техните клиенти са една автономна система. Възможно е в

някои случаи клиент на даден доставчик да бъде своя собствена автономна система, обикновено това се налага, когато клиентът е свързан към два или повече доставчика едновременно.

Автономните системи се означават чрез номер, който се раздава на организациите от регистраторите, които се грижат и за раздаване на IP адресите и са описани в глава 5. Старите номера на автономни системи са 16 битови и имат стойности от 1 до 65535. Подобно на IPv4 адресите те също са изчерпани, затова в момента номерата на автономните системи са 32 битови и се записват като две десетични числа със стойности от 1 до 65535, разделени с точка, например 5.1348. Някои стари маршрутизатори не могат да разпознават двуцифрените номера на автономни системи. Структурата на Интернет, разделен на автономни системи е показана на фигура 7.3.




**Фиг. 7.3. Автономни системи**

Външният свят гледа на автономната система като едно цяло. За него не е от значение за кой град или клиент е дадения пакет, целта е да се намери най-добрият път до автономната система. Веднъж влязъл вътре обаче приоритетите се променят и пакетът трябва да намери най-добрия път до конкретния компютър, клиент и град. Това означава, че целите в двата случая са различни и е необходимо да се обменя различна информация вътре в автономната система и между различните системи.

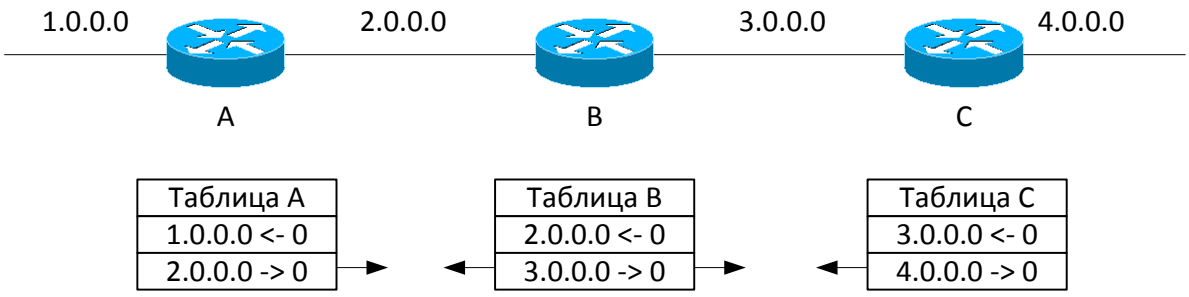
Затова са създадени два класа маршрутизиращи протоколи – външни (Exterior Gateway Protocols, EGP) и вътрешни (Interior Gateway Protocols, IGP), които работят между или вътре в автономните системи. В историята на Интернет е имало различни

външни протоколи, но в момента един е де-факто стандарт за външен протокол – това е BGP (Border Gateway Protocol).

Използването на BGP не е проста задача, този протокол изисква доста ресурси от маршрутизатора – например той трябва да има достатъчно памет, за да побере всички мрежи в Интернет, които в момента на писането на тази глава са около 500 000 само за IPv4. Използването на BGP обикновено е съпроводено с наличието на собствен номер на автономна система и собствено IP адресно пространство. Това е необходимо в следните случаи: ако сте Интернет доставчик от поне национално ниво, ако сте организация с две или повече връзки към различни Интернет доставчици или ако искате да бъдете разпознаван в Интернет отделно и независимо от вашия Интернет доставчик. Останалите протоколи, които ще бъдат разгледани в главата са вътрешни.

### 7.3.2 Протоколи, използваши вектор на разстоянието (Distance Vector)

При този клас протоколи всеки маршрутизатор предава цялата си таблица само на своите съседи. Това се прави на предварително дефиниран период от време, независимо дали има промяна в мрежовата топология. Пример за работата на този клас протоколи е показан на фигура 7.4.

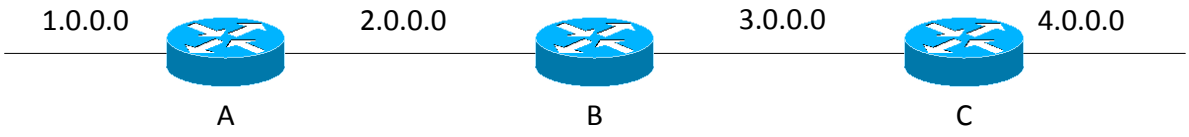



**Фиг. 7.4. Начално положение на мрежата**

В началото всеки маршрутизатор знае само за директно свързаните мрежи. В неговата таблица са записани тези мрежи, заедно с разстоянието (Distance) до тях – нула, защото са директно свързани и посоката (Vector), накъдето се достига до тях. Разбира се в истинските таблици вместо посоки наляво и надясно са записани интерфейсите на маршрутизатора, където са свързани мрежите. В този пример за опростяване като мярна единица за разстояние до мрежа се използва броят на преходите или броя маршрутизатори, през които трябва да премине пакет, за да се достигне до мрежата.

За да научат всички маршрутизатори за всички мрежи, всеки от тях предава таблицата на съседите си. Така B получава таблицата на A, игнорира записа за мрежа 2.0.0.0, защото тя вече присъства в неговата таблица и добавя нов запис за мрежа 1.0.0.0, която до този момент му е била неизвестна. За посока той записва интерфейсът, който е свързан към A, а за разстояние – към метриката 0, получена от съседа, той

добавя 1 – предварително известното разстояние между B и A и така полученият резултат 1 се записва в таблицата. По същия начин получавайки таблицата на C, B научава за мрежа 4.0.0.0 и я записва в таблицата си в посока през C на разстояние 1. Тази негова новопостроена таблица се предава към A и C, те научават за новите мрежи и се достига резултатът, показан на фиг. 7.5.




| Таблица A |
|---|
| 1.0.0.0 <- 0 |
| 2.0.0.0 -> 0 |
| 3.0.0.0 -> 1 |
| 4.0.0.0 -> 2 |

| Таблица B |
|---|
| 2.0.0.0 <- 0 |
| 3.0.0.0 -> 0 |
| 1.0.0.0 <- 1 |
| 4.0.0.0 -> 1 |

| Таблица C |
|---|
| 3.0.0.0 <- 0 |
| 4.0.0.0 -> 0 |
| 2.0.0.0 <- 1 |
| 1.0.0.0 <- 2 |

**Фиг. 7.5. Работно положение на мрежата**

В това положение всички маршрутизатори знаят за всички мрежи. То се нарича състояние на конвергенция и е работното състояние на мрежата. При промяна в топологията, например отпадане на мрежа 4.0.0.0 пръв за това разбира C, защото неговия интерфейс променя състоянието си на изключено. В зависимост от протокола той предава новата си таблица или веднага при научаване на промяната или при изтичане на таймера за предаване. Новата информация достига до B, той актуализира таблицата си и я предава към A. Така маршрутизаторите актуализират новата информация един по един, което в голяма мрежа може да отнеме доста време.

### 7.3.3 Протоколи, използващи състоянието на собствените си връзки (Link State)

При този клас протоколи всеки маршрутизатор предава на всички останали в дадения участък от мрежата само състоянието на собствените си връзки (свързаните към него мрежи). Така всеки получава информация от всички останали, в която се съдържат всички възможни пътища до всички мрежи. Тази информация всеки събира в база данни, след което стартира алгоритъма на най-късия път (Shortest Path First, SPF), който построява дървовидна структура на мрежата, подрежда дървото, като избира най-късите пътища и ги записва в маршрутизиращата таблица, като най-добри. За мрежите от предишния пример началното и крайното положение са същите, разликата е в това, че след стартиране на мрежата от начално положение всеки предава на всички едновременно и всички стартират процедура по изчисляване на топологията.

При промяна, например отпадане на мрежа 4.0.0.0 отново за това научава пръв маршрутизатор C, но тази промяна той предава веднага едновременно до B и A и всички едновременно научават за промяната и актуализират новата информация.

При този клас протоколи предаване се прави само при промяна в мрежовата топология, при стабилна мрежа не се предават служебни съобщения.

### 7.3.4 Сравнение на двата класа протоколи

Протоколите от типа вектор на разстоянието предават и получават информация само от своите съседи, затова при случайно или нарочно предаване на грешна информация те я приемат като вярна, защото нямат база за сравнение. При протоколите със собствените връзки всеки получава информация от всички, така ако някой получи грешна информация той има с какво да я сравни и евентуално да оцени, че тя наистина е грешна – по-трудно могат да бъдат излъгани.

При първия клас промените се обработват във всеки маршрутизатор, преди да бъдат предадени към следващия за обработка – научаването на промените отнема повече време, докато при втория промяната се предава незабавно до всички и всички едновременно я обработват, което ги прави по-подходящи за големи мрежи.

При протоколите, използващи вектор на разстоянието се предават таблиците дори и когато няма промяна на мрежата – те относително постоянно натоварват мрежата със служебни съобщения. При класа със собствените връзки в началото при включването на мрежата всеки се опитва да предава на всички, при което мрежата се натоварва извънредно. След постигане на съгласие обаче при стабилна мрежа те не предават информация и по този начин не натоварват мрежата със служебни пакети.

Една от най-важните разлики обаче, която е в полза на първия клас е свързана с натоварването и ресурсите, които се изискват от устройството. При вектора на разстоянието използваните изчисления са доста прости – маршрутизаторът получава от съседа информация, добавя към нея предварително известна стойност и я записва в таблица. Това прави тези алгоритми приложими и при най-евтините устройства с малки ресурси. При втория клас се събира база данни, обхожда се, построява се дърво, подрежда се и от него се изработва маршрутизираща таблица, затова те се вграждат обикновено в по-сложни и скъпи устройства и се използват за по-големи мрежи.

## 7.4 Конкретни реализации на маршрутизиращи протоколи

### 7.4.1 RIP (Routing Information Protocol)

Това е един от най-простите и ненатоварващи маршрутизаторите протоколи. Той е от класа Distance Vector и оценява трасетата по един критерий – брой на преходите. Така той избира като най-добър път най-късия, а не най-бързия път. При него всеки маршрутизатор изпраща цялата си таблица на съседите на всеки 30 секунди и има ограничение за максимален брой на преходите 15. Ако някой маршрут стане с метрика 16, съответният маршрутизатор обявява мрежата за недостижима. Затова той е подходящ за малки и прости мрежи. Фактът че е отворен стандарт обаче, в комбинация

с ниските изисквания за ресурси го правят много разпространен – той се поддържа от почти всички модели и марки маршрутизатори, включително и домашни такива, както и от повечето операционни системи, напр. Linux и Windows Server.

Протоколът RIP има три версии.

- RIP версия 1 е най-старата версия. Тя е класова и съди за размера на мрежата по нейния клас, без да предава мрежовата маска. Това може да създаде известни проблеми в съвременния безкласов Интернет. При тази версия съобщенията се предават като broadcast, което натоварва не само маршрутизаторите, а и останалите устройства в мрежата.
- RIP версия 2 е безкласова реализация, която предава мрежовата маска и я използва за изчисление на размера на мрежата. Освен това тя предава таблиците като Multicast, по този начин не оказва влияние на другите устройства в мрежата.
- RIPng (от New Generation) е създаден за маршрутизиране на IPv6 пакети.

### 7.4.2 OSPF (Open Shortest Path First)

Това е един сложен протокол от класа Link State. Неговите разработчици твърдят, че той може да работи успешно в мрежи, съставени от 1000 и повече маршрутизатора. При него оценката на трасетата се прави на базата на критерия цена (cost), която се изчислява автоматично от скоростта на трасетата, т.е. той се стреми да избира най-бързия път. Той е доста по-натоварващ за устройствата, затова въпреки че е отворен стандарт, обикновено се вгражда в по-високия клас устройства на повечето производители, има го реализиран в Linux и във Windows сървър до версия 2008.

Протоколът поддържа йерархично разделяне на мрежата на отделни области (areas), което позволява ограничаване на актуализациите при промени в мрежовата топология, например ако един клиент се включва или изключва в даден град, това да не се предава в останалите градове.

Този протокол предава информация при промяна на мрежовата топология, като използва Multicast адреси. Съвременните версии са безкласови, като предават и използват мрежовата маска. В момента се използват две версии:

- OSPFv2 – използва се за IPv4
- OSPFv3 – за IPv6

### 7.4.3 EIGRP (Enhanced Interior Gateway Routing Protocol)

Този протокол е разработен от Cisco Systems и до скоро беше тяхна собственост, което ограничаваше реализацията му само в устройства на тази фирма. В началото на 2013 г. Cisco Systems отвори протокола за общо използване, с цел да позволи реализацията му от други производители. До момента на написване на тази глава на

автора не са известни готови реализации от други производители, но много вероятно те скоро ще бъдат факт, поради очевидните предимства на протокола.

Според класификацията той се нарича подобрен Distance Vector – използва лесни изчисления, което го прави лек и с ниски хардуерни изисквания, в същото време изпраща актуализации само при промяна едновременно до всички, с което не натоварва мрежата и бързо актуализира промените. Така той комбинира предимствата на двата класа протоколи.

За определяне на най-добрия път EIGRP може да използва комбинация от четири критерия – скорост на линията (bandwidth), закъснение на пакетите (delay), надеждност на предаването (reliability) и натоварване на интерфейсите (load), което го прави най-гъвкав от всички протоколи по отношение на оценката. По подразбиране обаче използва само два – скоростта на линията и времезакъснението.

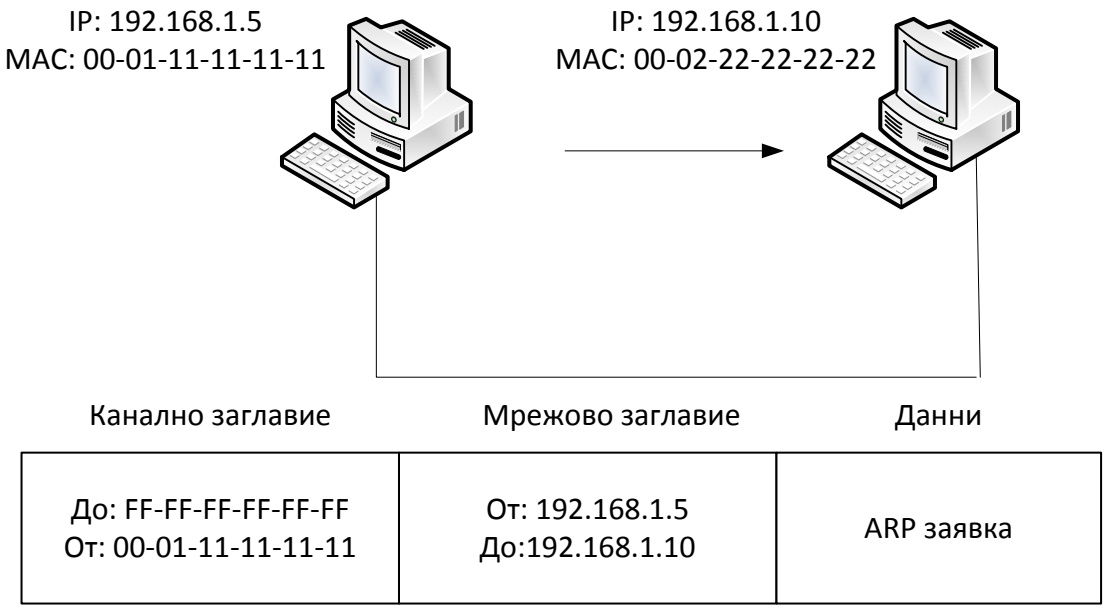
Протоколът има още едно основно предимство пред останалите при използването на няколко трасета едновременно за достигане на една и съща мрежа – техника позната още като балансиране на натоварването (Load Balancing). Останалите протоколи могат да използват няколко трасета едновременно, само ако те са с еднакви параметри или метрика – балансиране на натоварването по трасета с еднаква цена (Equal cost load balancing). Протоколът EIGRP е единственият, който може да балансира натоварването по трасета с различни параметри (Unequal cost load balancing), натоварвайки ги равномерно – например ако едното трасе е три пъти по-бързо от другото, протоколът ще изпрати три пакета по него и един пакет по второто трасе. По този начин се постига максимална скорост на обмен близка до получената при събиране на скоростта на отделните трасета.

# 8. Спомагателни протоколи на мрежово ниво

Освен маршрутизираните протоколи, като IPv4 и IPv6, които пренасят данните на потребителите и маршрутизиращите протоколи, с които мрежовите устройства си предават информация за мрежите, на мрежово ниво работят още няколко важни протокола, които осигуряват спомагателни функции на мрежата и на крайните устройства.

## 8.1 Протокол за съвпадения на адреси (Address Resolution Protocol, ARP).

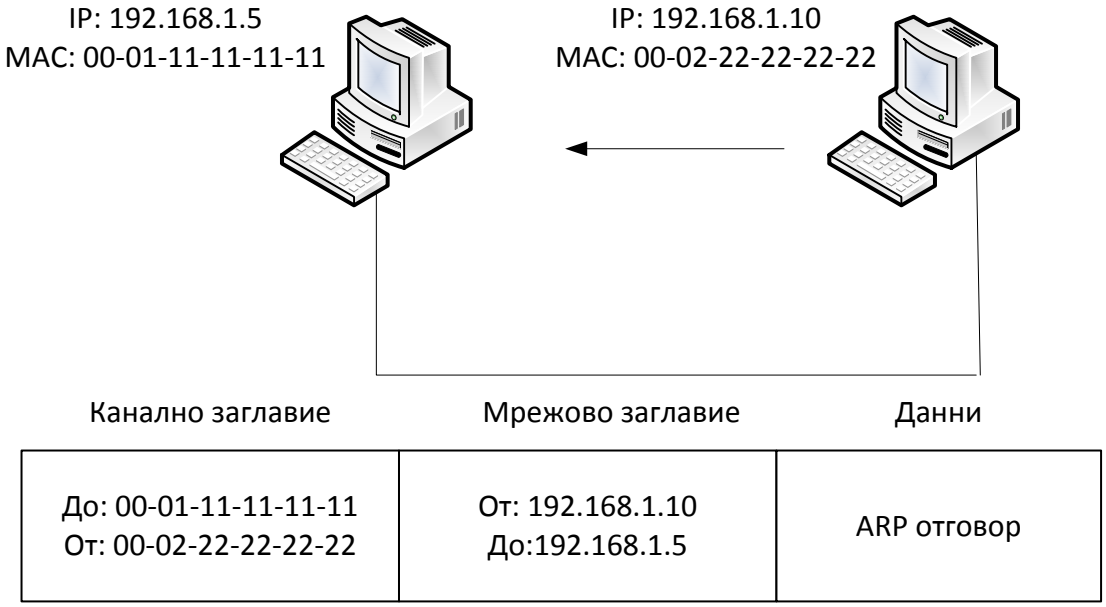
При описанията на начините за предаване на данни беше споменато, че когато компютърът трябва да предаде една порция информация на друг компютър в мрежата, в заглавната част на мрежовото ниво той трябва да сложи IP адреса на получателя, а в заглавната част на канално ниво – MAC адреса на получателя, ако той е в същата мрежа или MAC адреса на своя шлюз по подразбиране, ако получателят е в друга мрежа. IP адресът на кореспондента се научава обикновено на приложно ниво, например чрез услугата „Система за именуване на области“ (Domain Name System, DNS), описана подробно в последната глава. Тази услуга се грижи да получи IP адреса на съответния сървър от името на сайта, което потребителят изписва в адресното поле на уеб браузъра си. Умножавайки своя адрес и този на получателя по мрежовата маска, компютърът разбира дали на канално ниво трябва да постави MAC адреса на кореспондента си или на маршрутизатора, свързащ го към Интернет. След като бъде определено устройството, към което текущо ще бъдат предадени данните, източникът трябва да научи MAC адреса му, за да го запише като получател.




**Фиг. 8.1. Структура на ARP заявка.**

Автоматичното научаване на MAC адресите на другите устройства в мрежата става с помощта на протокола за съвпадения на адреси – ARP. За да научи MAC адрес на друг компютър в мрежата по познат IP адрес, източникът изпраща ARP заявка, чиято структура е показана на фигура 8.1.

В заглавната част на мрежовото ниво запитващият поставя своя IP адрес като източник и известният му IP адрес на получател, чиито MAC адрес търси. В заглавната част на каналното ниво поставя своя MAC адрес като източник, а за получател broadcast MAC адреса. В полето за данни се поставя идентификатор, че това е ARP заявка, тоест получателят трябва да отговори със своя MAC адрес. Пускайки така пакета по мрежата, той бива получен от всички компютри в дадения сегмент, защото в broadcast MAC адреса те припознават своя адрес. Всички получават данните, буферират ги и ги проверяват за грешки, след което полученият пакет се предава на мрежовото ниво. Там се проверява IP адреса на получателя и във възлите, за които няма съвпадение информацията се изхвърля без да се предприема допълнително действие. Само във възела, чиито IP адрес е записан като получател обработката продължава – той вижда че пакетът е за него, поглежда данните и разбира, че това е запитване за неговия MAC адрес. Затова той изготвя ARP отговор (reply), показан на фигура 8.2.

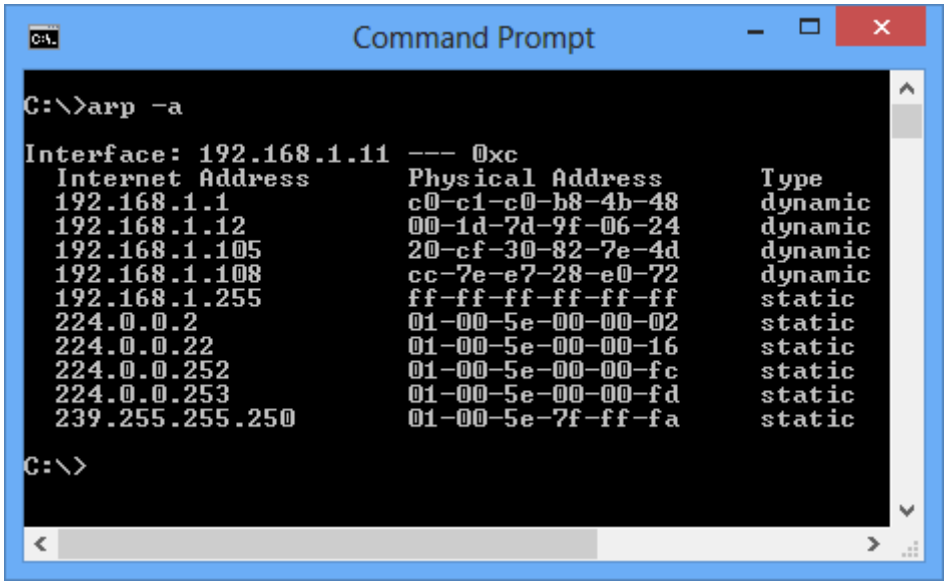



**Фиг. 8.2. Структура на ARP отговор.**

В двете заглавни части отговарящият поставя своите IP и MAC адреси за източник и тези на кореспондента си за получател. Така отговорът стига само до дадения получател, без да ангажира останалите компютри в мрежата[3].

[3] При някои операционни системи отговорът се изпраща също като broadcast, позволявайки на всички да научат съответствието, с цената на увеличаване на натоварването на устройствата и мрежата.

Описаният дотук алгоритъм позволява автоматичното научаване на MAC адресите на устройствата в мрежата, но с цената на broadcast предаване, което натоварва всички компютри в мрежата с обработката на ARP пакетите, които в частност интересуват само двете страни в комуникацията. За да може да се намали broadcast трафика всеки възел в мрежата съставя своя ARP таблица, в която записва познатите му съответствия между IP и MAC адреси. Когато на даден компютър му потрябва да научи MAC адреса на друг, той първо търси дали съществува такъв запис в таблицата. Ако съществува – взима MAC адреса от там, без да изпраща ARP заявка. Примерна структура на ARP таблица от операционна система Windows 8 е показана на фигура 8.3.




**Фиг. 8.3. ARP таблица.**

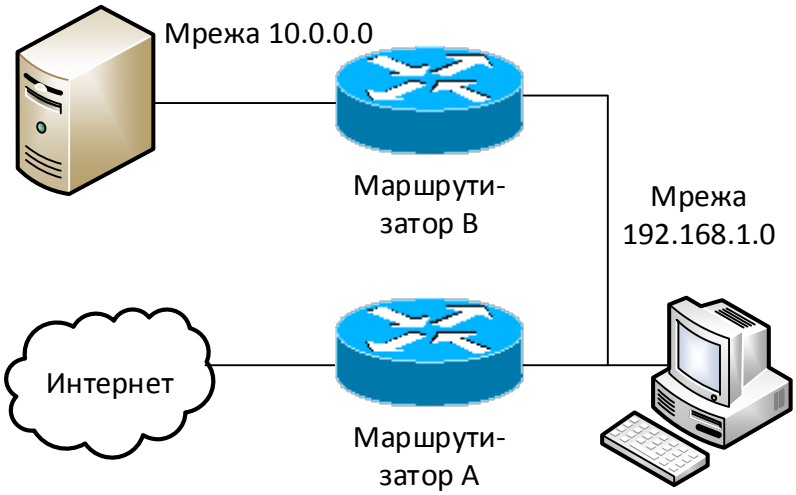
Таблицата съдържа познатите на компютъра IP адреси и съответните им MAC адреси, както и типа на записа – статичен или динамичен. Динамичните записи са тези, които са научени автоматично, благодарение на ARP протокола. Те обикновено се маркират по време и след изтичането на подразбиращото се време от няколко минути се изтриват от таблицата при липса на нова комуникация със съответния адрес. При някои операционни системи при изтичане на таймера редът не се изтрива, а се препотвърждава – изпраща се нова ARP заявка до същия адрес, за да се разбере дали съответствието още е валидно.

В таблицата могат да се правят и статични записи. Те се добавят ръчно от администратор на системата, при някои нови операционни системи, например Windows 8 има създадени няколко статични записа по подразбиране. Статичните записи не остаряват и не се премахват от таблицата автоматично, но могат да бъдат премахнати ръчно. Обикновено те служат за намаляване на broadcast трафика от динамичните ARP заявки, но могат да бъдат използвани и за защита – ако там напишем IP адреса на даден Интернет сайт и до него сложим невалиден MAC адрес или валиден такъв, но на някой друг компютър, то пакетите ще бъдат изпращани до записания в

таблицата MAC адрес и съответният сървър няма да може да бъде отварян на дадения компютър или ще може да се пренасочи към друга страница.

### 8.2 Вариация на протокола - ProxyARP

При тази вариация маршрутизаторът се настройва така, че да отговаря със собствения си MAC адрес за IP адрес на друга система. Пример за използването му е показан на фигура 8.4.




**Фиг. 8.4. Приложение на ProxyARP.**

В този пример връзката към Интернет преминава мрез маршрутизатор A, но има друга вътрешна мрежа, която е свързана към нашата през маршрутизатор B. Така според адресната схема, на компютъра трябва да се зададе за шлюз по подразбиране IP адреса на маршрутизатор A. По този начин той ще има връзка към Интернет, но няма да може да достига до мрежа 10.0.0.0, защото следвайки логиката на предаване ще бъде изчислено, че получателят е в друга мрежа. Така компютърът ще добавя в каналното ниво MAC адреса на маршрутизатор A и ще предава пакета в грешна посока.

Тази задача може да бъде решена и със средствата на маршрутизацията, добавяйки в маршрутизиращата таблица на компютъра статичен маршрут към мрежата, която трябва да достигаме по различен от подразбиращия се път. Съвременните компютри също имат маршрутна таблица и могат да позволят такова решение, но някои устройства, например мрежови принтери нямат възможност за подобна намеса. При тях проблемът може да бъде решен с протокола ProxyARP. Това означава на двата маршрутизатора да бъде стартиран този протокол, за да започнат те да отговарят със своите MAC адреси за получателите, за които предават пакетите и на компютъра да не бъде назначаван шлюз по подразбиране. Тогава за всеки получател, който не е от неговата мрежа компютърът ще изпраща ARP заявка до broadcast адрес, тя ще бъде получавана и от двата маршрутизатора и само този, който е отговорен за достигане на

дадения получател ще отговаря със своя MAC адрес, така пакетите ще достигат до правилните получатели.

Тази реализация е натоварваща за маршрутизаторите, поради повечето дейности, който трябва да изпълняват и за мрежите, поради увеличения брой broadcast пакети и рядко се използва. Повечето смислени употреби на ProxyARP в момента са при доставчиците на Интернет услуги, за да ограничават използването на свободните IP адреси или да не позволят на потребителите си да си предават информация директно.

### 8.3 Протокол за управляващи съобщения в Интернет (Internet Control Message Protocol, ICMP)

Този протокол се използва за две цели – изпращане на служебни съобщения за осигуряване на допълнителни функции, например диагностика на мрежовата връзка и за изпращане на съобщения за възникнали грешки по време на движението на пакет. Тъй като различните видове пакети се използват за различни цели, съобщенията имат в заглавната си част число, наречено „тип на съобщението“, определящо за какво ще се използва. Част от основните типове съобщения са показани в таблица 8.1.

**Табл. 8.1. Типове ICMP съобщения**

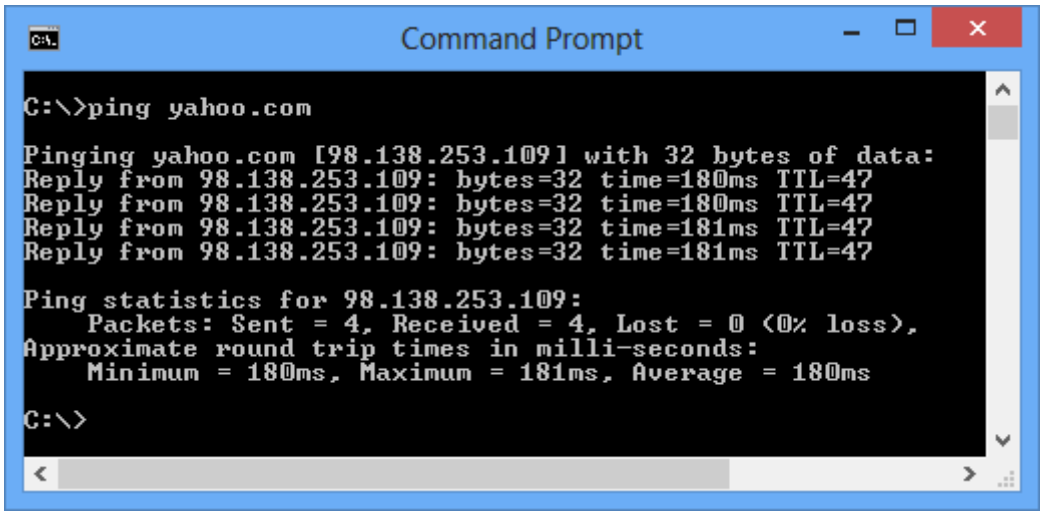
| Тип | Значение |
|---|---|
| 0 | Echo reply (Отговор на ехо заявка) |
| 3 | Destination Unreachable (Не мога да стигна до получателя) |
| 5 | Redirect (Пренасочване) |
| 8 | Echo Request (заявка за ехо) |
| 9 | Router Advertisement (Обява от маршрутизатор) |
| 10 | Router Solicitation (Търсене на маршрутизатор) |
| 11 | Time Exceeded (Изтекло време) |
| 12 | Parameter Problem: Bad IP header (Проблем с параметър: грешка в заглавието) |
| 13 | Timestamp (Маркер за време) |
| 14 | Timestamp Reply (Отговор на маркер за време) |

Освен показаните в таблицата типове съобщения съществуват и други, които са се използвали за различни цели в миналото, но сега техните функции се изпълняват от други протоколи, затова те са поставени в статус „изоставен“ (Deprecated).

Някои видове съобщения нямат различни варианти, а други имат. За разграничаване на различните причини например за невъзможност за достигане до получателя, ICMP пакетите имат и код на съобщението. При тези, при които няма вариации, кодът е със стойност 0, за останалите различният код означава различни причини за грешката или варианти на командата. Следва обяснение на някои от функциите, изпълнявани от протокола.

### 8.3.1 Проверка за достижимостта до даден възел (Ping)

Проверката дали пакетите достигат до даден възел и се връщат успешно обратно се изпълнява благодарение на ICMP съобщенията Echo Request и Echo Reply. Чрез програмата Ping, която е част от всяка съвременна операционна система проверяващият изпраща няколко ICMP Echo Request пакета, които в нормална ситуация достигат до получателя. За всеки от тях получателят изработва Echo Reply пакети, които връща обратно. Когато отговорът достигне до програмата, тя изписва на екрана времето, изтекло от изпращането на запитването до получаването на отговора. Примерен изход от програмата Ping е показан на фигура 8.5.




**Фиг. 8.5. Резултат от Ping.**

В примера се проверява дали пакетите достигат успешно до Интернет сайта yahoo.com и обратно. За да проверим това в командния терминал изписваме командата: ping yahoo.com. На първия ред компютърът ни научава IP адреса на сайта yahoo.com, след което изпраща подред четири ICMP Echo request пакета с размер 32 байта до този адрес. Интернет сайтът ги получава, връща отговорите чрез ICMP Echo Reply пакети, при получаването на всеки от които нашия компютър изписва по един от четирите реда по-долу, които казват: „Получен е отговор от адрес 98.138.253.109 с размер 32 байта, времето от изпращането на запитване до получаването на отговор е 180 милисекунди, времето на живот (TTL) на получения пакет е 47. Отдолу се изписва статистика за броя изпратени пакети, успешно получени отговори, минимално, максимално и средно време за получаване на отговорите.

Програмата Ping може да бъде стартирана с различни опции, които да променят броя изпращани пакети, размера им, времето за изчакване на отговор и други параметри. Това я прави доста полезен инструмент за мрежова диагностика. В операционната система Microsoft Windows програмата изчаква получаването на отговор или изтичането на времето за изчакване (timeout) и тогава изпраща следващия пакет. При операционните системи Linux е възможно да я накараме да изпрати

следващия пакет без да изчаква отговора на предишния. Това означава, че с помощта на тази програма бихме могли да изпратим голям брой пакети към дадена система или мрежа, с което да навредим на работоспособността и. Затова някои системи имат вградени защити срещу такива пакети – например не отговарят изобщо на Echo Request или ограничава броя отговори, за да се защитят срещу атаки. Това означава, че липсата на отговор от дадена система не означава гарантирано, че тя или мрежата до нея не работят правилно.

### 8.3.2 Съобщение за грешка „Не мога да стигна до получателя“ (Destination Unreachable)

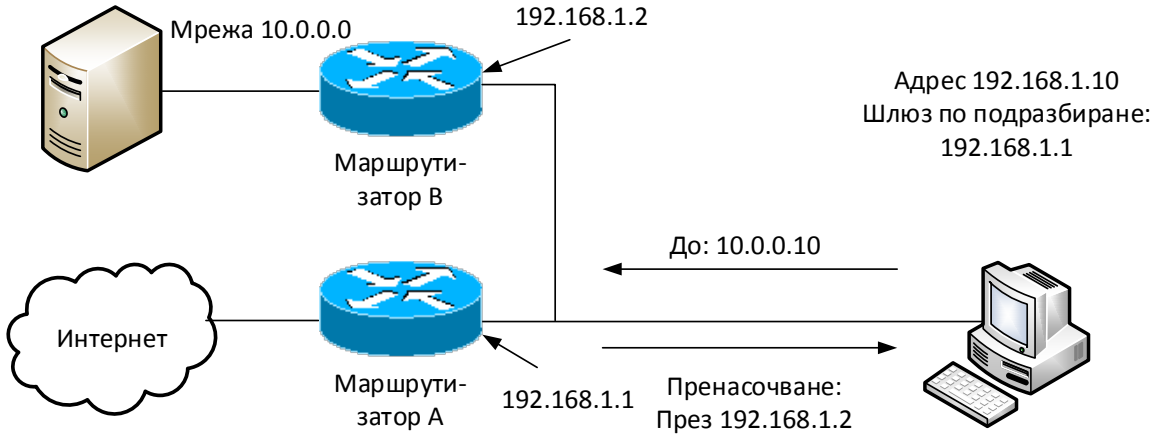
Както беше обяснено, пакетът преминава през различни маршрутизатори по пътя си от източника до получателя и всеки от тях прави своя обработка на пакета, решавайки накъде да го изпрати. Във всеки такъв междинен възел може да се случи по някаква причина той да не може да изпрати пакета по-нататък. Тогава той изхвърля пакета и връща ICMP съобщение на източника Destination Unreachable. Понеже причините за невъзможност за достигане до даден получател са различни, това съобщение има и код, показващ причината, поради която не може да се достигне получателя. Типични кодове на съобщението са показани в таблица 8.2.

**Табл. 8.2. Кодове на ICMP съобщението „Destination Unreachable”.**

| Код | Значение |
|---|---|
| 0 | Destination network unreachable (Не мога да стигна до мрежата на получателя) |
| 1 | Destination host unreachable (Не мога да стигна до крайния получател) |
| 2 | Destination protocol unreachable (Използваният протокол не е достъпен) |
| 3 | Destination port unreachable (Не мога да предам до указания номер на порт) |
| 4 | Fragmentation required, and DF flag set (Необходима е фрагментация, но тя е забранена от източника) |
| 5 | Source route failed (Маршрутизацията от източника е неправилна) |
| 6 | Destination network unknown (Непозната мрежа на получателя) |
| 7 | Destination host unknown (Непознат получател) |
| 8 | Source Host Isolated (Източникът е изолиран) – вече не се използва |
| 9 | Network administratively prohibited (Мрежата на получателя е забранена) |
| 10 | Host administratively prohibited (Достъпът до получателя е забранен) |
| 11 | The network is unreachable for Type Of Service (Типът на услугата е забранен за мрежата на получателя) |
| 12 | The host is unreachable for Type Of Service (Типът на услугата е забранен за получателя) |
| 13 | Communication Administratively Prohibited Комуникацията е забранена) |
| 14 | Host Precedence Violation (Нарушение за приоритета до получателя) |
| 15 | Precedence cutoff in effect (По-нисък приоритет от допустимия) |

### 8.3.3 Пренасочване (Redirect)

Показаната на фигура 8.4 мрежа, при която Интернет се достига през един маршрутизатор, а друга мрежа се достига през втори маршрутизатор беше решена в точка 8.2 чрез използването на протокола ProxyARP, но използването му е нежелано, поради допълнителното натоварване върху мрежата и устройствата. Същата задача може да бъде решена с помощта на протокола ICMP и неговото съобщение „пренасочване“ (redirect). Диалогът е показан на фигура 8.6.




**Фиг. 8.6. Съобщение „пренасочване“ (Redirect).**

Тук за шлюз по подразбиране (Default Gateway) на компютъра е настроен IP адресът на маршрутизатор A (192.168.1.1). Така той предава всички пакети към него и когато получателят е в Интернет, маршрутизатор A успешно обработва пакета. Когато обаче се изпраща пакет за получател от мрежа 10.0.0.0, която се достига чрез маршрутизатор B, компютърът който не знае това го изпраща отново до A. Той изхвърля пакета и връща ICMP съобщение със смисъл: „Пренасочи този пакет през маршрутизатор B“. Кодовете на съобщението са показани в таблица 8.3.

**Табл. 8.3. Кодове на ICMP съобщението „Redirect”.**

| Код | Значение |
|---|---|
| 0 | Redirect Datagram for the Network (Пренасочи за цялата мрежа) |
| 1 | Redirect Datagram for the Host (Пренасочи за този получател) |
| 2 | Redirect Datagram for the TOS & network (Пренасочи за този тип на услугата за цялата мрежа) |
| 3 | Redirect Datagram for the TOS & host (Пренасочи за този тип на услугата и конкретния получател) |

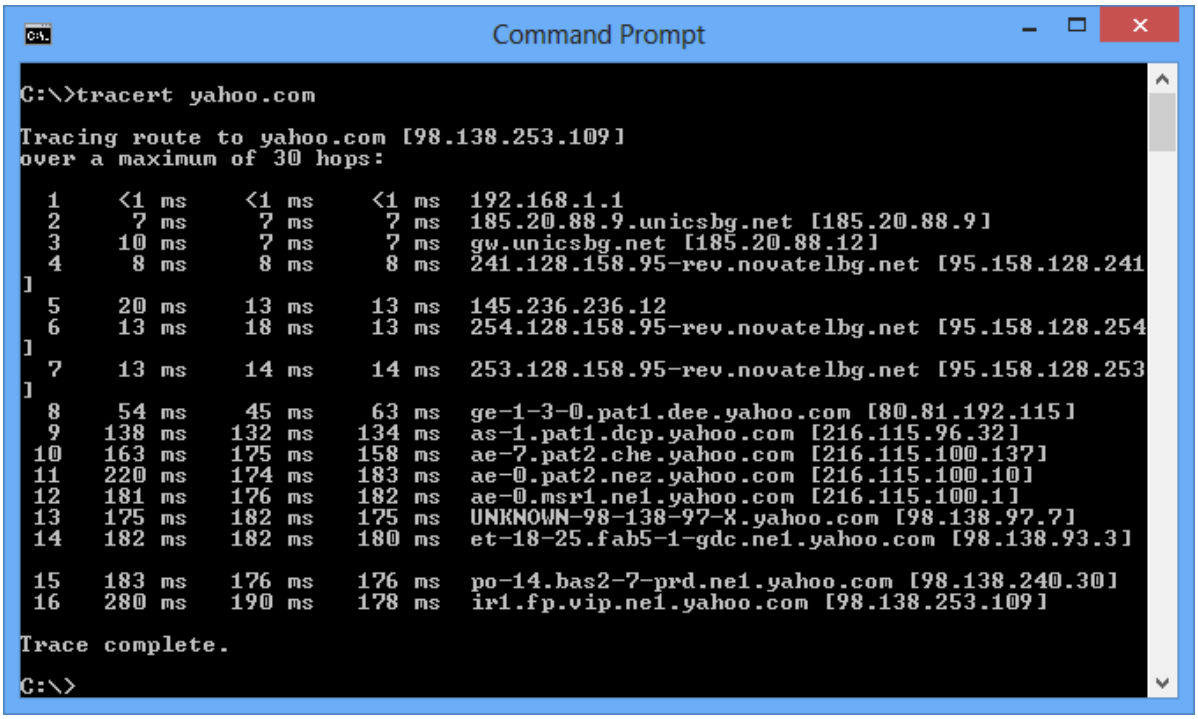
### 8.3.4 Съобщения „Търсене на маршрутизатор“ и „Обява за маршрутизатор“

Тези съобщения са незадължителни при IPv4. Те могат да се използват по следния начин: маршрутизаторът може през определено време, например няколко десетки минути да изпраща в мрежата пакет „Обява за маршрутизатор“ (Router Advertisement), чрез което компютрите в мрежата автоматично да научават за свързаните маршрутизатори и да могат да ги използват. Компютрите в мрежата могат при нужда да изпращат пакети „Търсене на маршрутизатор“ (Router solicitation), на които маршрутизаторите да отговарят. Тези две съобщения съставят т. нар. „протокол за откриване на маршрутизатори в Интернет (Internet Router Discovery Protocol, IRDP), който не е добре развит и използван при IPv4, но се използва при IPv6.

### 8.3.5 Изтекло време (Time Exceeded)

При преминаване на пакет през маршрутизатор, той намалява времето на живот (TTL) в заглавната част на пакета. Маршрутизаторът, който намали времето на живот и то стане 0 изхвърля пакета и връща ICMP съобщение на източника Time Exceeded.

Това поведение се използва от друга диагностична програма – traceroute (под Windows се нарича tracert), която се използва за проследяване на пътя, по който минават пакетите, за да достигнат до получателя. Примерен изход от програмата под Windows 8 е показан на фигура 8.7.




**Фиг. 8.7. Изход от програмата Traceroute.**

Програмата изпраща три последователни пакета до указания получател с TTL=1. Пакетът стига до първия маршрутизатор, който намалява TTL, то става 0, пакетът се

изхвърля и се изпраща ICMP съобщение на получателя, където програмата Traceroute получава съобщението за грешка, изчислява изминалото време от изпращането на пакета до получаването на грешката и изписва резултата в трите колони след номера на възела за всеки от трите пакета. След това изпраща пакети с TTL=2. В първия маршрутизатор TTL става 1, във втория 0 и така се разбира кой е втория по пътя. Действието се повтаря с увеличаване на TTL, докато се достигне до получателя. В примера на фигурата са показани всички 16 стъпки от източника до получателя.

### 8.4 Протокол ICMPv6

За целите на мрежите, работещи с IPv6 протокол е разработен и протоколът ICMPv6, който изпраща диагностични пакети и съобщения за грешки. Неговото действие е подобно на това на предишния, но някои типове съобщения са премахнати, а са добавени нови, осигуряващи допълнителни функции.

С помощта на някои нови типове съобщения в ICMPv6 е разработен протоколът за откриване на съседи (Neighbor Discovery Protocol, NDP). Той разчита на следните съобщения:

- Обява за маршрутизатор (Router Advertisement) – съобщение, което се изпраща периодично от маршрутизатора или при запитване от краен възел;
- Търсене на маршрутизатор (Router Solicitation) – изпраща се от крайния възел, за да намери маршрутизатора, обслужващ мрежата.
- Обява за съсед (Neighbor Advertisement) – изпраща се периодично или като отговор на запитване, за установяване на адресите на съседните възли;
- Търсене на съсед (Neighbor Solicitation) – изпраща се от един възел към друг, с цел научаване на неговия адрес или други параметри.

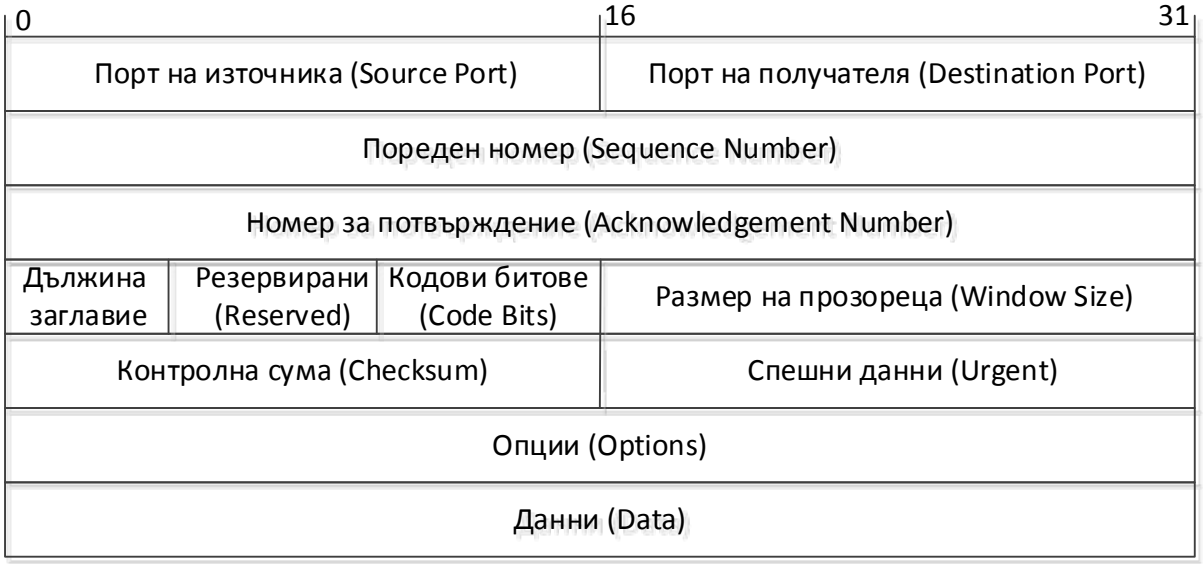
Благодарение на първите две съобщения се осигурява автоматичното конфигуриране на адреси, описано в глава 6, а вторите две съобщения осигуряват функции, подобни на протокола ARP, описан в точка 8.1. Предимството на механизмите при IPv6 е, че при комуникациите се използват многоцелеви (multicast) пакети, вместо broadcast, както при IPv4 и така пакетът се получава и обработва само от възлите, за които е предназначен, тоест намалява се натоварването на мрежата.

# 9. Транспортно ниво

Транспортното ниво в OSI модела се грижи за надеждното предаване на информация. В Интернет обаче на това ниво може да работи един от двата протокола – TCP, който осигурява надеждност или UDP, който не се грижи за надеждното предаване на данни. Характеристиките и функциите на двата протокола са описани в тази глава.

## 9.1 Протокол за управление на предаването (Transmission Control Protocol, TCP).

Заглавната част на протокола TCP е показана на фигура 9.1.




**Фиг. 9.1. Заглавна част на TCP протокол.**

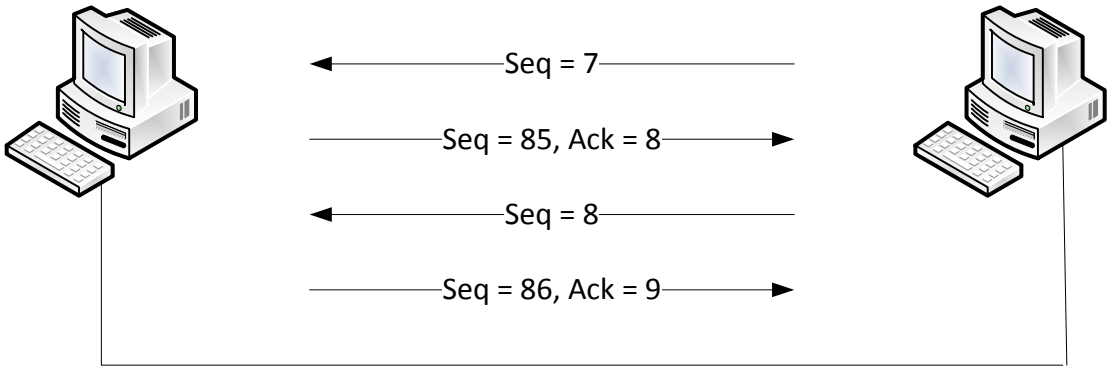
Полетата в заглавната част са:

- Порт на източника (Source Port) – шестнадесет-битово поле, в което се записва номерът на порта, идентифициращ програмата на източника, изпращаща данните;
- Порт на получателя (Destination Port) – шестнадесет-битово поле, в което се записва номерът на порта, идентифициращ програмата на получателя, за която са предназначени данните;
- Пореден номер (Sequence Number) – тридесет и две битово поле, указващо поредния номер на изпращаните данни;
- Номер за потвърждение (Acknowledgement Number) - тридесет и две битово поле, указващо кой пореден номер данни се потвърждават като получени;
- Дължина на заглавната част (Header Length или Data Offset) – число, указващо дължината на заглавната част на протокола в 32 битови порции. Минималната дължина е 20 байта, което се указва с число 5 в това поле;

- Резервирани (Reserved) – шест бита за бъдещи разширения. В някои разработки последните три от тях имат назначен смисъл, но те са още в експериментален статус;
- Кодови битове (Code Bits) – шест бита, използвани за установяване, поддържане, прекратяване на съединения и за потвърждения;
- Размер на прозореца (Window Size) – шестнадесет-битово поле, използвано за динамично регулиране на количеството данни, които да се изпращат и потвърждават наведнъж;
- Контролна сума (Checksum) – шестнадесет-битово поле, служещо за проверка за грешки. В него се записва сумата на всички останали двубайтови полета от заглавната част и данните;
- Спешни данни (Urgent) – Рядко използван механизъм, който позволява част от данните да бъдат маркирани като спешни и приемането им да се сигнализира незабавно на потребителя, без да се изчаква приемане на целия ресурс;
- Опции (Options) – незадължителни допълнителни функции на протокола. Най-често използваната опция е „Максимален размер на сегмента“ (Maximum Segment Size), определяща размера на данните, които страните ще предават наведнъж.
- Данни (Data) – поле, съдържащо предаваните данни.

### 9.1.1 Потвърждения при TCP протокол

Тъй като основна функция на протокола TCP е да осигури правилното предаване на данните, при него се обменят потвърждения като индикатор, че дадена порция данни е получена правилно. Използват се двете полета от заглавната част – пореден номер (Sequence Number, Seq) и номер за потвърждение (Acknowledge Number, Ack). Процесът е показан на фигура 9.2.

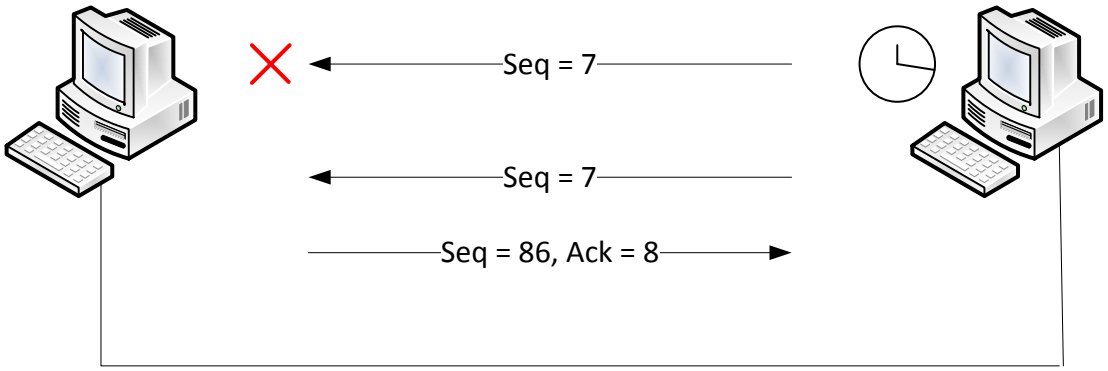



**Фиг. 9.2. Потвърждения при TCP протокол.**

Полето „Пореден номер” (Seq) има стойност между 0 и 4 294 967 295, която се увеличава за всеки предаден пакет, а при достигане на максималната стойност започва

отново от нула. Практически увеличението на поредния номер във всеки следващ пакет е с броя на предадените байтове в полето за данни, но за опростяване на фигурата е представено увеличаване с 1 – броя предадени пакети. Източникът вдясно на фигурата предава своя пакет с пореден номер 7. При правилно получаване на данните, приемникът вляво отговаря със стойност в полето „пореден номер“ (Seq) със своя пореден номер, до който е стигнал в момента – в примера 85, а в полето „Номер за потвърждение“ (Ack) той записва стойност 8, което означава „получил съм правилно номер 7, сега очаквам да ми изпратиш номер 8.

Потвърждения се изпращат само при правилно получена информация. При грешна информация или неполучен пакет приемникът не изпраща нищо. Такъв пример е показан на фигура 9.3.




**Фиг. 9.3. Диалог при неполучена или грешна информация.**

При всяко предаване източникът стартира таймер, предава данните, оставя ги в буфер в паметта си и чака за потвърждение. При получаване на потвърждение за тези данни източникът ги премахва от буфера и предава следващата порция данни, както е показано на фигура 9.2. Ако до изтичането на таймера не се получи потвърждение, източникът изпраща отново данните.

Колко време се изчаква преди повторното изпращане или колко пъти се прави опит за повторно изпращане са настройки на конкретната операционна система. Например при Microsoft Windows те се записват в базата данни, наречена Registry.

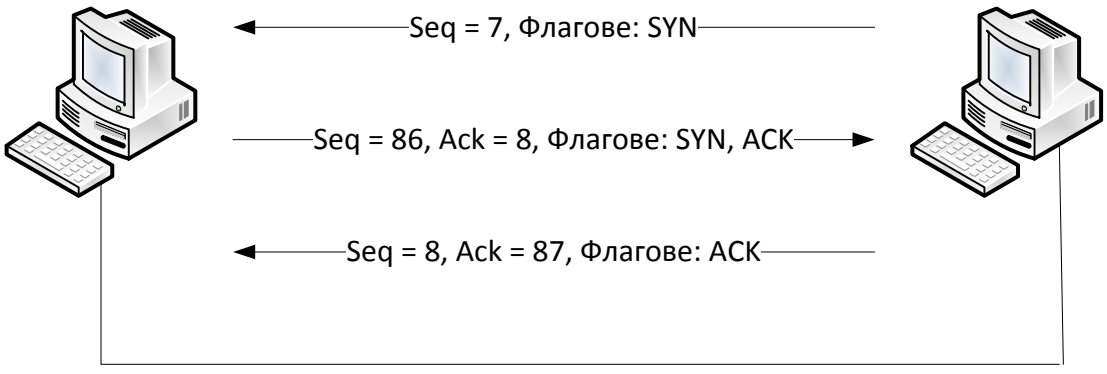
### 9.1.2 Кодови битове

Шестте кодови бита от полето служат за установяване, поддържане и разпадане на TCP съединение (Connection) между двата компютъра, които си предават данни. Самият начин на установяване на съединението е показан в следващата точка, тук се описва значението на самите кодови битове. Те са описани според начина на подреждането им в заглавната част.

- URG (Urgent) – указва че стойността, записана в полето от заглавната част „Urgent” е валидна и трябва да бъде проверена от получателя.
- ACK (Acknowledge) - указва че стойността, записана в полето от заглавната част „Acknowledgement Number” е валидна и трябва да бъде проверена от получателя. На практика указва, че с този пакет се потвърждава правилно получен друг пакет с номер, записан в полето „номер за потвърждение“.
- PSH (Push) – Указва данните да не се задържат в буфер до получаването на целия ресурс (файл), а да се предадат веднага към програмата, която ги очаква.
- RST (Reset) – служи за аварийно прекратяване на връзката при загуба на комуникацията в едната посока.
- SYN (Synchronization) – служи за установяване на връзка. Вдигнат е в 1, когато текущия източник иска да установи връзка с отсрещния кореспондент.
- FIN (Finalize) – служи за нормално прекратяване на връзката, когато всички данни вече са предадени.

### 9.1.3 Установяване и прекратяване на връзка

Преди да могат да се предават данни в която и да е посока, между двете страни на комуникацията при TCP протокол трябва да се установи връзка (TCP Connection). При установяването на връзката двете страни обменят поредния си номер, който ще използват при комуникацията и определят някои други параметри, например максималния размер на сегмента и размера на прозореца, описани в следващите точки. Въпреки че не е задължително, в практиката обикновено инициаторът за установяване на връзката е клиентът (например web browser), а отговарящият е сървърът. Установяването на връзката се осъществява с обмена на три сегмента и на английски се нарича Three-way-handshake. Процесът е показан на фигура 9.4.

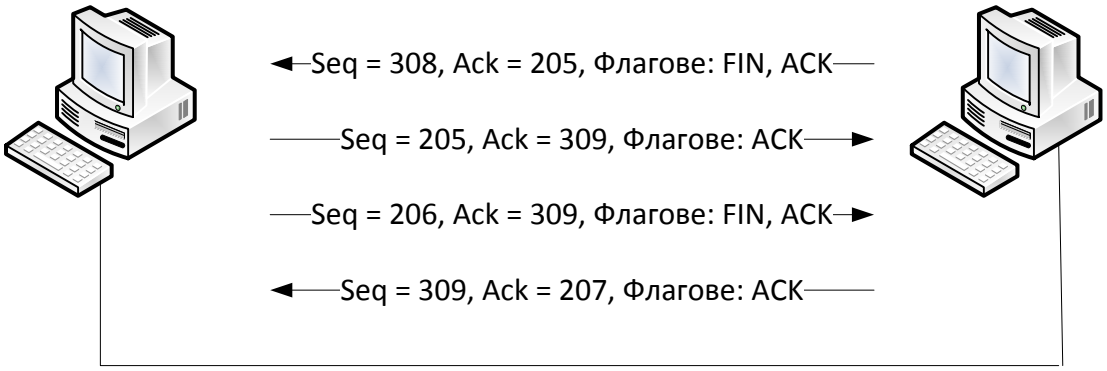



**Фиг. 9.4. Установяване на TCP съединение.**

Клиентът вдясно на фигурата изпраща сегмент с вдигнат кодов бит SYN, което е покана за осъществяване на връзка. В полето „Пореден номер“ (Seq) той изработва и записва случайно число, от което ще стартира броенето за тази връзка, в примера 7.

Ако сървърът също иска да установи връзка, изпраща обратно втори пакет с вдигнат бит SYN, което е положителен отговор на поканата за осъществяване на връзка. Освен него е вдигнат и бита за потвърждение ACK, с което указва, че потвърждава успешното получаване на пакета с номер 7 и очаква като следващ да получи номер 8 (стойността, записана в полето „Номер за потвърждение“ – Ack). В полето „пореден номер“ той изпраща своята случайна стойност, в примера 86. Процесът завършва с изпращането на третия пакет от клиента към сървъра – той е с вдигнат бит ACK, за да потвърди успешното получаване на сегмент номер 86. Двете страни вече знаят поредните номера, които ще се изпращат. От тук нататък може да започне обмен на данни във която и да е от двете посоки или дори и в двете едновременно.

Всяка от страните може да предложи прекратяване на връзката – например клиентът е получил необходимия ресурс (web страница) и може да затвори връзката или сървърът да е отчел достигане на максимално позволен ресурс (напр. време). Процесът за нормално прекратяване на връзка е показан на фигура 9.5.




**Фиг. 9.5. Прекратяване на TCP връзка**

Страната, която предлага прекратяването на връзката изпраща сегмент с вдигнати битове ACK и FIN. Другата страна потвърждава получаването на предложението с един сегмент, след което изпраща втори сегмент с вдигнати битове ACK и FIN. Четвъртият сегмент потвърждава прекратяването на връзката.

### 9.1.4 Управление на потока данни

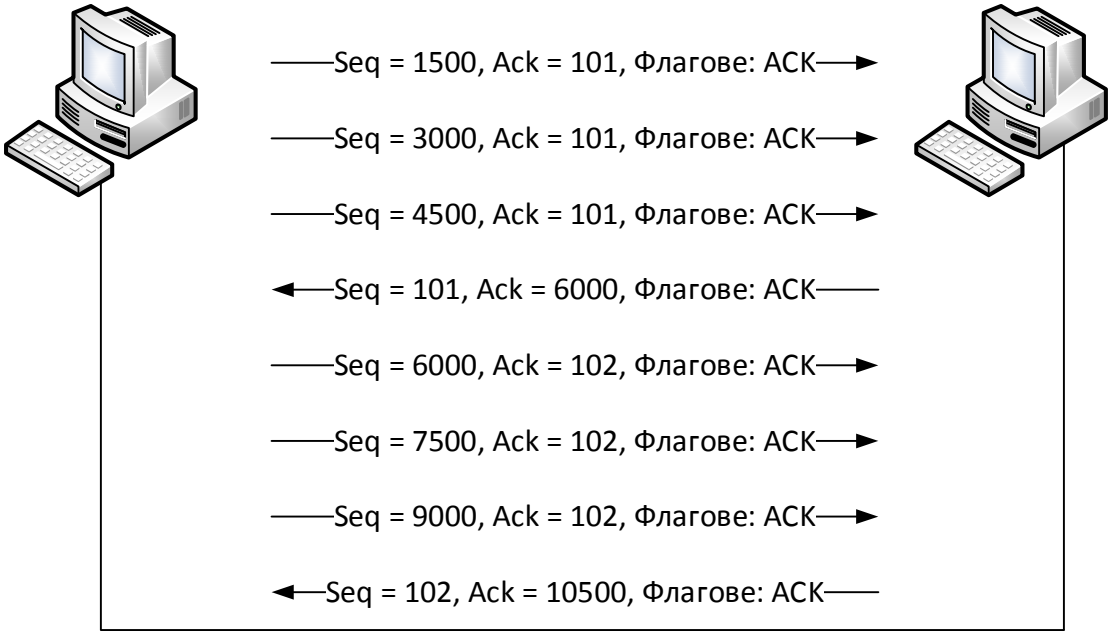
Управлението на потока данни има за цел да регулира скоростта на предаване между източника и получателя и да определи количеството данни, които ще се предават и потвърждава наведнъж. Механизмът при TCP се нарича „Плъзгащ се прозорец“ (Sliding Window). Размерът на прозореца означава броя байтове, които източникът може да предаде преди да спре и да очаква потвърждение. Тъй като той може да се променя динамично по време на вече установена връзка се нарича „плъзгащ се“. Един прост механизъм за потвърждения може да се представи, като се

определи, че всяка порция данни трябва да бъде потвърдена. Такъв механизъм е показан на фигура 9.6.


**Фиг. 9.6. Прост механизъм за потвърждения.**

В примера се предполага, че от сървъра вляво към клиента вдясно се предават по 1500 байта данни наведнъж, а клиентът изпраща само потвърждения. Затова поредният номер на сегментите на сървъра се увеличава с по 1500 байта, а този на клиента с по един. При простия механизъм за потвърждения всеки сегмент трябва да бъде потвърден. Така сървърът изпраща една порция данни и чака за потвърждение, преди да изпрати следващата. Този начин не изисква сложно управление, но не е много ефективен при всякакви среди за предаване, особено при несиметрични трасета, при които скоростта на предаване в едната посока може да надвишава десетки пъти скоростта в обратна посока.

За да може предаването да бъде по-ефективно може да се увеличи количеството предавани данни, което да бъде потвърждавано с един сегмент. Ако в примера на фигура 9.5 размерът на прозореца е 1500 байта (един сегмент), може да се предположи, че сървърът може да предава по 3 сегмента, преди да чака потвърждение или това съответства на размер на прозореца 4500 байта. Така сървърът предава три последователни сегмента с поредни номера 1500, 3000 и 4500 и те биват потвърдени наведнъж със следващия сегмент с поле „Номер за потвърждение 6000, т.е. очаквам номер 6000. Следващият прозорец, който се предава се състои от трите сегмента с поредни номера 6000, 7500 и 9000 и те биват потвърдени наведнъж със сегмент с номер за потвърждение 10500. Това предаване при липса на грешки е показано на фигура 9.7.

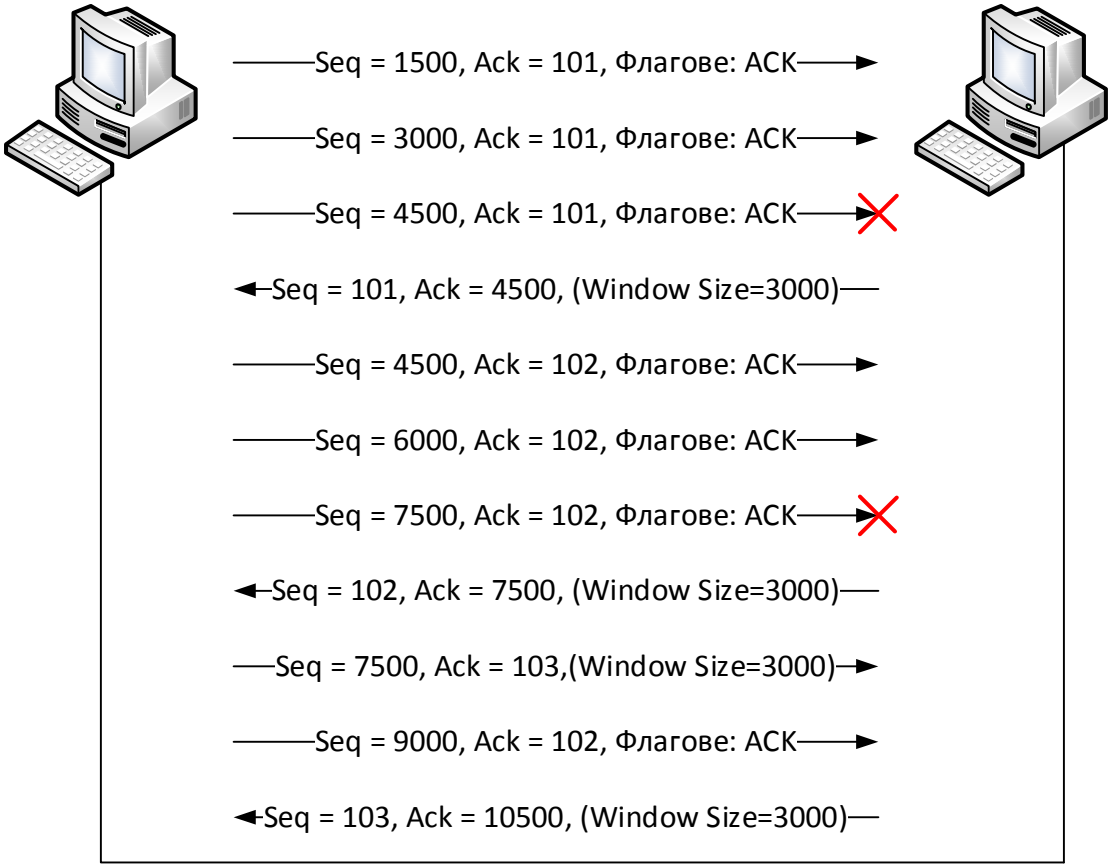



**Фиг. 9.7. Предаване при размер на прозореца 4500.**

Прозорецът може да се настройва динамично по време на вече установена връзка. Ако едната страна предава по повече данни, отколкото другата може да обработи (например няма достатъчно буферно пространство да запази толкова данни), може да се достигне ситуацията, показана на фиг. 9.8.

Тук източникът предава с размер на прозореца 4500 байта (3 сегмента), но приемникът има възможност да обработва само два сегмента (3000 байта). Така той приема първите два, изхвърля третия и изпраща сегмент за потвърждение за правилно получаване на първите 3000 байта (Ack=4500 означава „Сега очаквам да получа сегментът с байтове до 4500). Това потвърждение може да съдържа в себе си предложение за намаляване на размера на прозореца на 3000 байта, защото приемникът може да получава само по толкова. Източникът може да приеме веднага това предложение, а може и да продължи да предава по 4500 байта наведнъж, както в примера. От номера за потвърждение източникът разбира, че байтове до 3000 са получени успешно, а от 3001 до 4500 не са получени. Затова той предава следващия прозорец – три сегмента по 1500 байта, съответно с поредни номера 4500, 6000 и 7500. Приемникът отново обработва първите два, изхвърля третия и връща потвърждение Ack=7500, което е сигнал, че сега очаква да получи сегментът, завършващ на байт 7500.

В този пример след поредното неуспешно приемане на третия сегмент източникът предполага, че приемникът не може да приема по три наведнъж, защото той винаги потвърждава по два успешно получени и намалява размера на прозореца на два сегмента или 3000 байта.




**Фиг. 9.8. Намаляване на размера на прозореца.**

Възможно е и динамично увеличаване на размера на прозореца, с цел по-ефективно използване на връзката. Ако няколко пъти подред се предадат успешно определено количество данни е възможно едната страна да предложи увеличаване на прозореца и да започнат да се предават по повече наведнъж.

### 9.1.5 Едновременно предаване (мултиплексиране) на различни потоци данни

Съвременните компютърни и операционни системи са многозадачни и могат да работят с няколко мрежови програми едновременно. Възможно е в даден момент от време едновременно да се изпраща или получава електронна поща, да се отваря една или няколко уеб-страници, да се изтегля файл и да се изпращат съобщения. Това означава че в някакъв интервал от време до нашия компютър пристига един пакет от електронната поща, един от файла, един от уеб-страницата и т.н. Разграничаването на отделните порции информация и изпращането им към съответните програми, които трябва да ги получат е задача на транспортното ниво и се осъществява с помощта на номерата на портовете.

В заглавната част на транспортните протоколи TCP и UDP има номер на порт на източника и номер на порт на получателя. Те определят програмите в двете компютърни системи, които изпращат и съответно трябва да получат тази порция

данни. Номерът на порта е 16 битово число и може да заема стойности от 0 до 65535. Разпределението на номерата е показано в таблица 9.1. [1]

**Табл. 9.1. Разпределение на номера на портове**

| | |
|---|---|
| 0 | Неназначен номер на порт |
| 1 - 1023 | Системни (добре познати) номера на портове |
| 1024 - 49151 | Потребителски (регистрирани) номера на портове |
| 49152-65535 | Динамични и/или частни номера на портове |

Порт номер 0 е резервиран. Обикновено когато дадено приложение показва, че използва този порт, това означава, че на него не му е назначен номер на порт.

Портове от 1 до 1024 се наричат системни или добре познати. Обикновено на тях работят сървърните приложения и по номера на порта може да се познае (или предположи) кое е приложението. В повечето реализации тези номера са запазени и не бива да се използват от други програми, въпреки че понякога това се прави, например популярният комуникатор Skype може да използва порт номер 80, който е запазен за Web сървъри. В миналото портовете за сървърни приложения са били в обхвата 1 – 255 и в някои литературни източници те се наричат „за публични приложения“ (Public Applications). На тях работят повечето стандартни и дълго използвани Интернет приложения, като web-сървъри, електронна поща и други. Обхватът 256 – 1023 се нарича още портове за пазарни приложения (Marketable Applications) и в него работят приложения, разработени доста след създаването на Интернет, например Adobe Flash.

Потребителските или регистрирани номера на портове могат да бъдат запазвани за определени програми. За разпределението на номерата на портовете се грижи организацията IANA (Internet Authority Numbers Association). Доста приложения имат запазени номера на портове в този диапазон, например Yahoo! Messenger използва порт номер 5050. Тези номера на портове често се използват в клиентските компютри за динамично назначаване на отделните програми от операционната система, когато не са заети от друга програма.

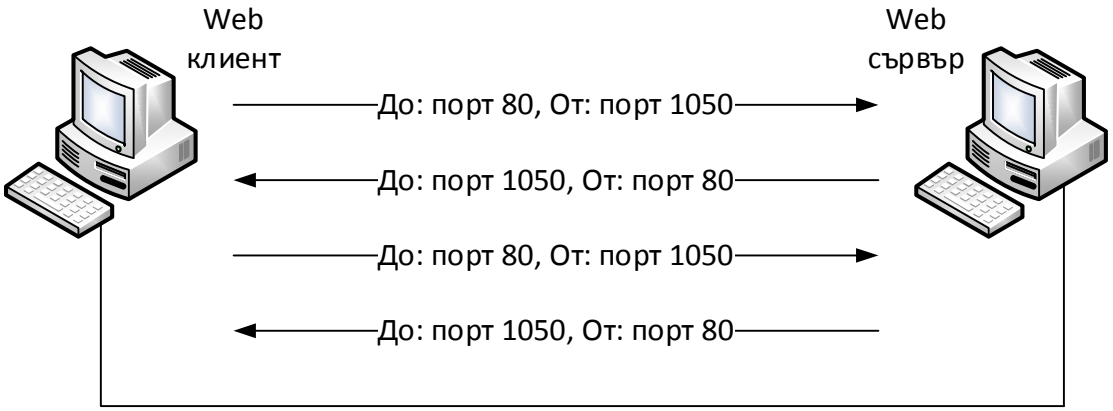
Динамичните или частни номера на портове не могат да бъдат регистрирани в IANA. Те се използват за специфични или временни цели. Някои маршрутизатори и защитни стени филтрират тези номера на портове по подобие на частните IP адреси и така те могат да се използват само в конкретна локална мрежа, не и в Интернет.

Въпреки че дадено приложение може да използва за транспорт само протокол TCP или UDP, когато за него има запазен номер на порт обикновено той е запазен и в двата протокола. В таблица 9.2 са показани запазените номера на портове на някои типични Интернет приложения.

**Табл. 9.2. Запазени номера на портове**

| Порт | Протокол | Приложение |
|---|---|---|
| 20 | TCP | Протокол за предаване на файлове (FTP) – порт за данни |
| 21 | TCP | Протокол за предаване на файлове (FTP) – порт за команди |
| 22 | TCP | Криптиран протокол за отдалечен достъп Secure Shell (SSH) |
| 23 | TCP | Некриптиран протокол за отдалечен достъп Telnet |
| 25 | TCP | Протокол за предаване на електронна поща Simple Mail Transfer Protocol (SMTP) |
| 53 | TCP/UDP | Протокол за именуване на области - Domain Name System (DNS) |
| 67/68 | UDP | Протокол за динамично конфигуриране на адреси – Dynamic Host Configuration Protocol (DHCP) |
| 69 | UDP | Прост протокол за предаване на файлове - Trivial File Transfer Protocol (TFTP) |
| 80 | TCP | Протокол за предаване на хипертекст - Hypertext Transfer Protocol (HTTP) |
| 110 | TCP | Протокол за получаване на електронна поща - Post Office Protocol v3 (POP3) |
| 123 | UDP | Протокол за синхронизация на време Network Time Protocol (NTP) |
| 137-139 | TCP/UDP | Споделяне на файлове при Microsoft Windows |
| 143 | TCP | Протокол за получаване на електронна поща – Internet Message Access Protocol (IMAP) |
| 161 | UDP | Протокол за мрежово наблюдение и управление - Simple Network Management Protocol (SNMP) |
| 179 | TCP | Маршрутизиращ протокол BGP (Border Gateway Protocol) |
| 443 | TCP | Криптиран HTTP протокол – Secure HTTP (HTTPS) |
| 445 | TCP | Споделяне на файлове при Microsoft Windows (SMB) |
| 520 | UDP | Маршрутизиращ протокол Routing Information Protocol (RIP) |
| 521 | UDP | Маршрутизиращ протокол за IPv6 - RIPng |
| 530 | TCP/UDP | Отдалечено извикване на процедури – Remote Procedure Call (RPC) |

Обикновено сървърното приложение заема своя запазен номер на порт и очаква комуникация на него от клиентите (например web сървърът очаква клиентите на порт номер 80). Когато стартираме клиентска програма, например web браузър, той иска свой номер на порт от операционната система и получава свободен такъв. Тъй като отделните прозорци (панели) на браузърите обикновено отварят различни сайтове, всеки от тях си има собствен номер на порт, за да може да получава пакетите от своя сайт. Примерна комуникация е показана на фигура 9.9.

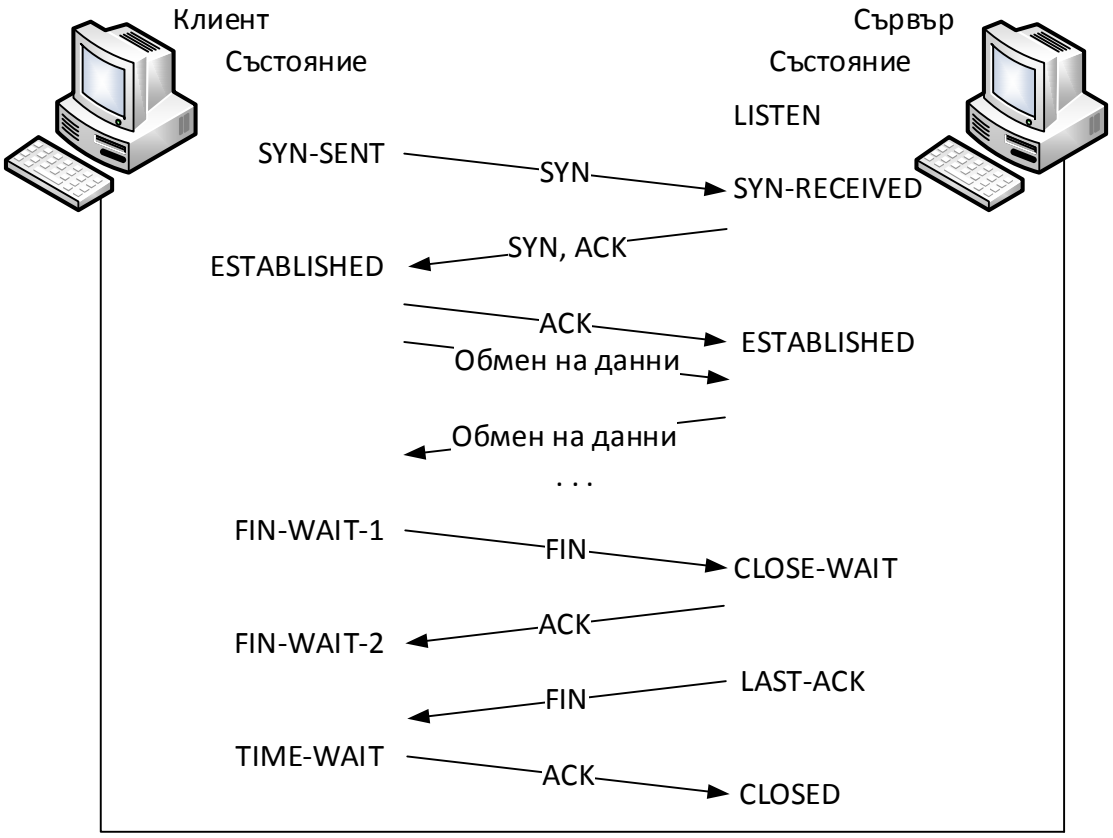



**Фиг. 9.9. Комуникация чрез портове.**

Състоянията на даден порт за конкретна операционна система могат да бъдат:

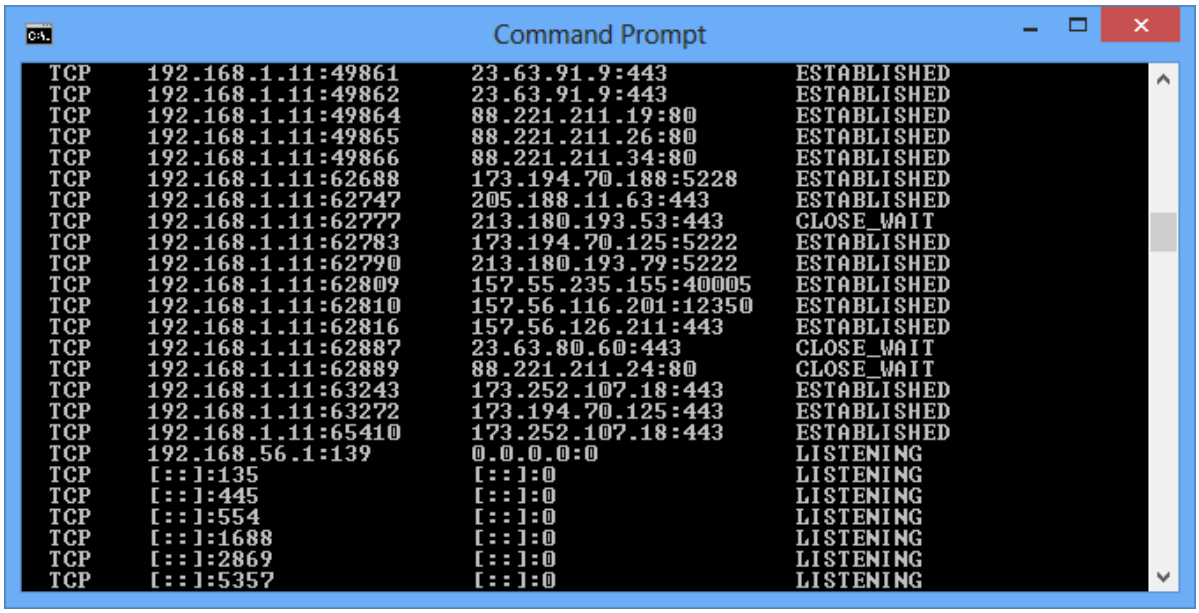
- LISTEN (в сървъра): очаква се заявка за връзка от отдалечен компютър;
- SYN-SENT (в клиента): Изпратена е заявка за връзка и се очаква отговор;
- SYN-RECEIVED (в сървъра): Получена е заявка за връзка;
- ESTABLISHED (в клиента и сървъра): връзката е установена – нормалното състояние за предаване на данни;
- FIN-WAIT-1 (в клиента и сървъра): Изпратена е заявка за прекратяване на връзката и се очаква отговор;
- FIN-WAIT-2 (в клиента и сървъра): Получено е потвърждение за прекратяване на връзката и се очаква предложение за прекратяване от отсрещната страна;
- CLOSE-WAIT (в клиента и сървъра): Получена е заявка за прекратяване на връзката от кореспондента и се очаква потвърждение от локалното приложение;
- CLOSING (в клиента и сървъра): Изчакване за потвърждение на изпратена заявка за прекратяване на връзката;
- LAST-ACK (в клиента и сървъра): Изпратено е последно потвърждение за прекратяване на връзката и се очаква потвърждение за получаването му;
- TIME-WAIT (в клиента и сървъра): Изчакване на определено време след изпращането на последното потвърждение, за да сме сигурни, че то е получено;
- CLOSED (в клиента и сървъра): Не е установена връзка.

Диаграма на установяване и прекратяване на връзка, заедно с обозначените състояния е показана на фиг. 9.10.




**Фиг. 9.10. Състояния на TCP протокол.**

Инструменти, с които може да се проверяват състоянията на портовете на даден компютър са netstat (от команден режим) при Windows и Linux, както и графичният безплатен инструмент за Windows – TCPview. Частичен примерен изход от командата netstat под Windows 8 е показан на фигура 9.11.




**Фиг. 9.11. Примерен изход от netstat.**

## 9.2 Протокол за потребителски дейтаграми UDP (User Datagram Protocol)

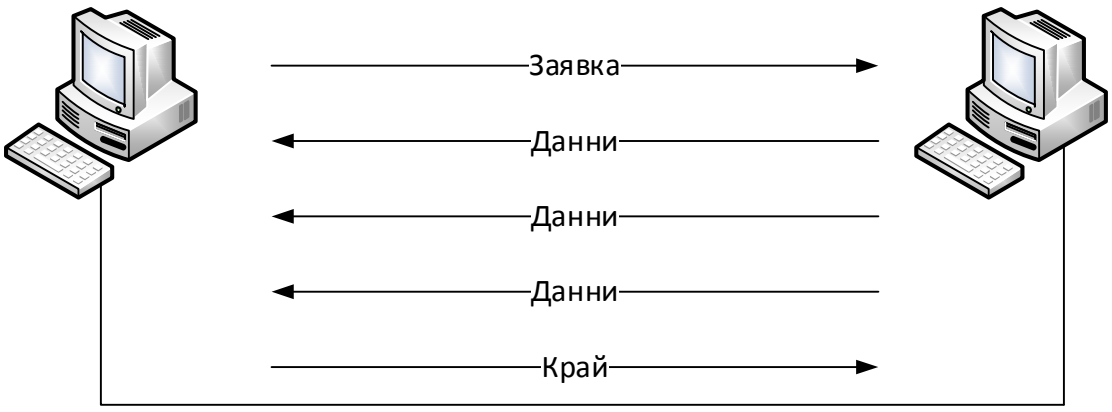
В терминологията на Интернет понятието „Дейтаграма“ (Datagram) означава “Самостоятелна независима единица данни, носеща достатъчно информация, за да бъде маршрутизирана от източника до получателя, без да разчита на предишен обмен между двата кореспондента и транспортната мрежа“.[3]

Заглавната част на протокола UDP е показана на фигура 9.12.


**Фиг. 9.12. Заглавна част на протокол UDP.**

Както се вижда, при този протокол няма полета за потвърждения, кодови битове за установяване и прекратяване на връзка, нито поле за регулиране на скоростта на предаване. Затова тези функции, осигуряващи надеждност отсъстват при протокола UDP. Предаването на данни при него се извършва по следния начин: когато даден клиент иска определен ресурс (например файл или поток от Интернет телевизия), той без да установява връзка изпраща към сървъра заявка. Сървърът разделя ресурса на отделни сегменти, ако той е по-голям и не може да се изпрати наведнъж, след това без да номерира сегментите ги изпраща подред към клиента. Клиентът получава поредица от сегменти, които събира в ресурс по реда на пристигането им и ги предава към приложението, без да изпраща потвърждения към сървъра.

Примерен обмен на данни при UDP протокол е показан на фигура 9.13.




**Фиг. 9.13. Примерен обмен на данни при протокол UDP.**

Разбира се липсата на пореден номер и потвърждения означава, че при този обмен липсва надеждност. Ако например даден сегмент не пристигне в получателя или пристигне с грешка, протоколът не осигурява повторното предаване на липсващите и сгрешените данни. Така приемникът получава всички данни без липсващите, събира ги в един ресурс и ги предава на приложението. Възможно е сегментите да пристигнат в разбъркан ред при получателя и това също няма как да бъде установено поради липсата на номерация. Обикновено при такъв обмен клиентът проверява дали получения ресурс е използваем и ако установи някаква грешка той изисква повторното предаване от сървъра.

Не винаги когато се използва за пренос протокол UDP липсва надежност при комуникацията. Възможно е да има механизъм за потвърждения и обработка на грешки, но той се реализира в самото приложение, обменящо данните и не се осигурява от транспортния протокол.

Протоколът UDP се използва широко при мултимедийни предавания на данни – Интернет телевизия, радио, IP телефония и др., защото при тях скоростта на предаване на данните е важна и закъсненията, породени от потвърждения и изчаквания при TCP протокола могат да се окажат твърде големи, за да може да се осигури услуга в реално време. Друг пример за използването на UDP комуникация са услугите от тип „заявка-отговор“, при които данните, които се предават са с малък размер и се събират в един сегмент. Пример за такава комуникация е протоколът DNS (Domain Name System), описан подробно в глава 11, който е предназначен да връща IP адреса на даден Интернет сървър (сайт) от неговото име, например клиентът трябва да установи връзка със сайта www.tugab.bg и трябва да получи неговия IP адрес, за което той изпраща заявка към специален DNS сървър. Ако се използва протокол TCP за транспорт, комуникацията би се получила с обмяната на три сегмента за установяване на връзка, три сегмента са получаване на информация (заявка, отговор и потвърждение) и четири сегмента за прекратяване на връзка – общо 10 сегмента. При използване на протокол UDP в нормална ситуация са необходими само два сегмента – заявка и отговор.

Заглавната част на протокола е само 8 байта, в сравнение с 20-те байта на TCP, което означава, че той предава по-малко служебна информация и по-малко натоварва мрежовите ресурси.

# 10. Сеансово и представително ниво

Сеансовото и представителното ниво на OSI модела отсъстват в TCP/IP модела. Това не означава, че в Интернет тези функции не могат да се реализират, а че те се изпълняват от приложното ниво, тоест по желание се реализират в приложението, което обменя данни. Характеристиките и функциите на двете нива са описани в тази глава.

## 10.1 Сеансово ниво (Session Layer).

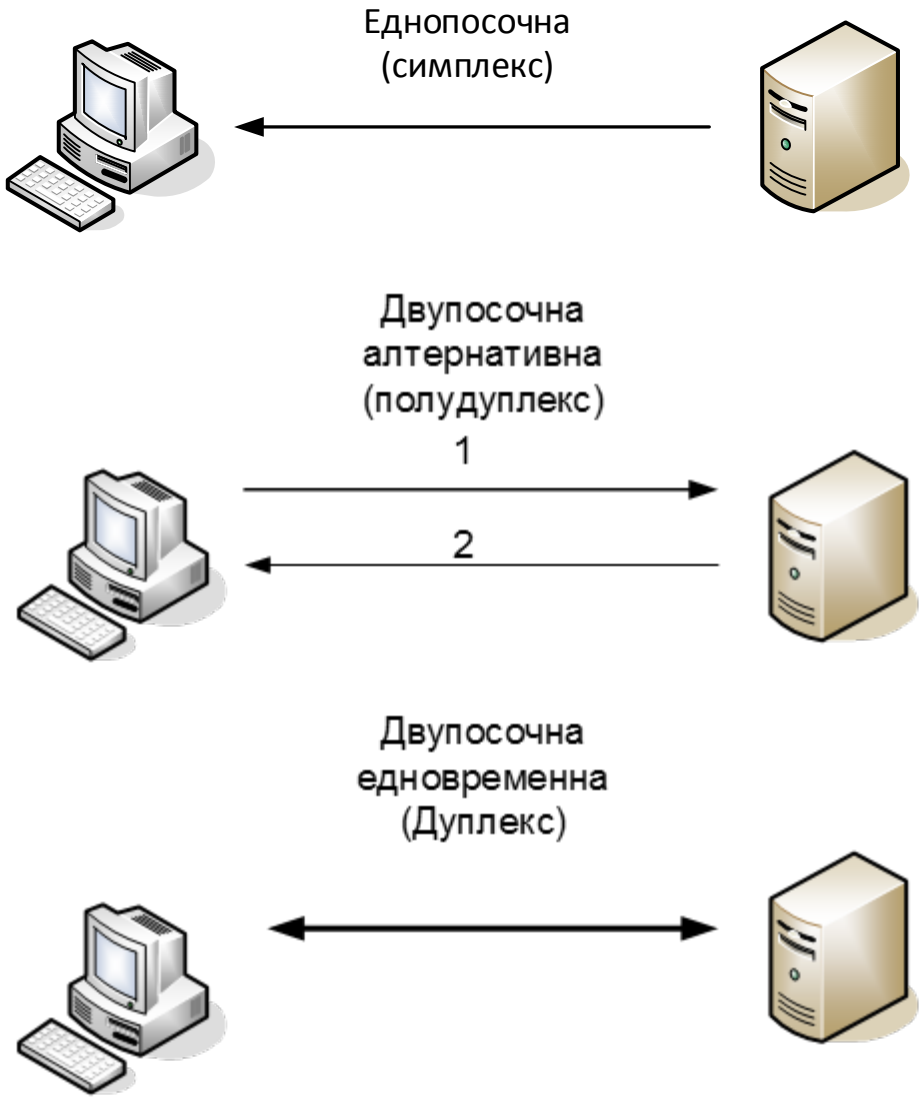
Сеансовото ниво се грижи за установяване, поддържане и разпадане на приложни съединения между програмите, които обменят данни. Приложните съединения се различават от транспортните съединения при протокола TCP, описани в предишната глава, въпреки че в някои случаи двете съединения могат да съвпадат по продължителност и функции. Транспортното съединение има за цел да предаде надеждно даден ресурс (файл, уеб-страница) между двата кореспондента, докато приложното съединение управлява диалога между двете приложения (клиент и сървър) по време на цялата сесия. Ако например имаме Web-сървър, който изисква оторизация на потребителите (вход с потребителско име и парола), функционалността за установяване на сесията на потребителя и проверка на името и паролата е част от функцията установяване на приложното съединение на сеансовото ниво. При вече установено приложно съединение, в зависимост от правата на потребителя в дадения сайт за него стават достъпни определени ресурси (файлове, уеб страници). При заявка за получаването на определен ресурс се установява ново транспортно съединение, ресурсът се предава до потребителя, транспортното съединение се прекратява, но сесията на потребителя остава установена до настъпване на някакво събитие – ръчно излизане на потребителя или достигане на максимално време на неактивност. Възможно е за ускоряване на комуникацията в рамките на дадена сесия да се установят няколко едновременни транспортни съединения – например ако потребителят заяви получаване на дадена web-страница, която се състои от няколко файла (текст, картинки), за всеки файл да се установи отделно транспортно съединение. Някои основни функции на сеансовото ниво са:

### 10.1.1 Установяване на диалога

Тази функция включва установяването на сесията. Тя може да съдържа в себе си проверката за валидно потребителско име и парола, наричана още „автентикация“ (authentication) и даването на достъп до ресурсите, до които потребителят има право, наричана още „оторизация“ (authorization). Обикновено тя включва и определянето на правилата за предаване на данни, посоката за обмен на данни, времето за валидност и/или неактивност на сесията и други параметри, определящи параметрите на услугата.

### 10.1.2 Управление на диалога

Функцията се грижи за изпълнение на правилата, договорени по време на установяването на диалога – кой ще предава пръв, в каква посока се обменят данните и други параметри. Например може да имаме еднопосочна (симплекс) комуникация, ако услугата е Интернет телевизия – данните се предават само от сървъра към клиента. Възможно е да има двупосочна алтернативна комуникация (полудуплекс), при която данни се предават и в двете посоки, но не едновременно – например при свързване със сървър за електронна поща първо да се получат писмата, които се намират в пощенската кутия на сървъра към клиента и после да се изпратят писмата, които са буферирани в клиента към сървъра. Възможно е да има и двупосочна едновременна комуникация (пълен дуплекс), например при телефонен разговор през Интернет. Примери за видове комуникация са показани на фигура 10.1.




**Фиг. 10.1. Видове комуникация.**

### 10.1.3 Синхронизация

Функцията синхронизация включва възможността за разделянето на комуникацията на по-малки елементи, които да бъдат потвърждавани или повторно предавани поотделно. Например ако предаваме група файлове последователно или поредица от електронни пощи, след всеки изцяло предаден елемент (файл, поща) е възможно да се изисква потвърждение за успешно получаване, което може да води до изтриване на ресурса от сървъра. При неправилно получаване е възможно да се изпраща сигнал за повторно предаване на целия ресурс.

Възможно е даден ресурс да бъде променен на сървъра, след като потребителят вече го е получил, например потребител, чиито профил в социална мрежа разглеждаме да смени профилната си снимка. Тогава чрез функцията синхронизация сървърът може да сигнализира на клиента, че е необходимо повторното прочитане на дадения ресурс и опресняването на информацията да стане автоматично.

### 10.1.4 Управление на активността

Управлението на активността може да включва следене и отчитане на използваните ресурси (време, обем на информацията), прекъсване на сесията след изчерпване на договорените ресурси или определен период на неактивност и др. Често тази функция се нарича отчитане (accounting). Обикновено в сървъра се пази отчет (log) за действията на всеки потребител по време на неговата сесия и този отчет може да бъде преглеждан при нужда.

### 10.1.5 Обработка на изключения

Обработката на изключения дава възможност за определяне на сигнализация и реакция при специфични грешки за дадения обмен на данни – например искаме да изпратим електронна поща на даден потребител, но неговата пощенска кутия е пълна и не може да получи даденото съобщение или искаме да запишем файл в папка, в която нямаме права за запис.

### 10.1.6 Протоколи на сеансово ниво

Част от протоколите на сеансово ниво са:

- Отдалечено извикване на процедури (Remote Procedure Call, RPC) – мрежова услуга, позволяваща изграждането на разпределени приложения. Обикновено клиентът изпраща данни към сървъра, извиквайки негова процедура, сървърът извършва определено действие с данните и връща резултата на клиента;

- Мрежова файлова система (Network File System, NFS) – разработен от SUN Microsystems и широко използван в UNIX/Linux операционните системи протокол за

отдалечен достъп до файлове, намиращи се на дисково устройство, разположено на друг компютър през мрежата;

- Система за структурирани запитвания (Structured Query Language, SQL) – широко използван механизъм за достъп до информация, намираща се в централизирана база данни на мрежов сървър.

## 10.2 Представително ниво (Presentation Layer)

Представителното ниво има за цел да представи информацията по подходящ начин, така че тя да бъде разбираема за двете системи, които си обменят данни. Тази функционалност е все по-важна в съвременните комуникации, когато имаме многообразие от програми и операционни системи (Windows, Linux, Android, MAC iOS и др.) и искаме да осигуряваме мрежови услуги, съвместими със всички клиенти. Нивото изпълнява три основни функции:

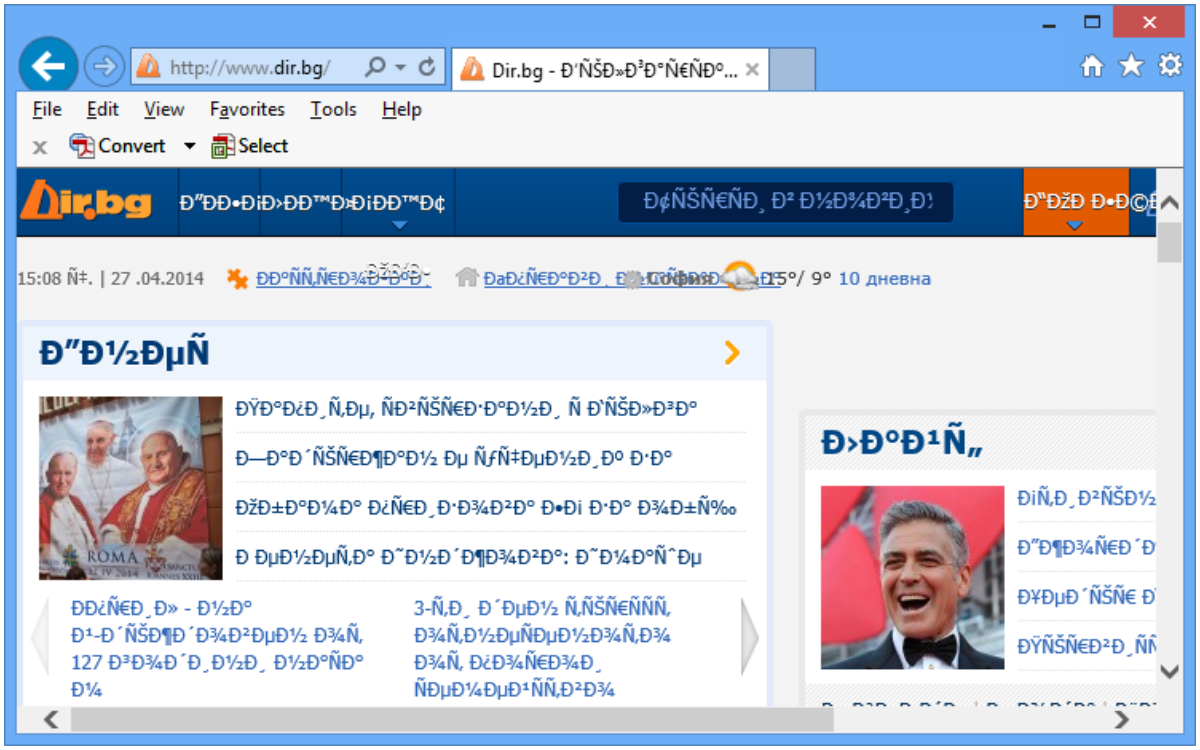
### 10.2.1 Форматиране (представяне) на данни

Форматирането на данни означава представянето или конвертирането на данните така, че те да имат еднакво значение в системата източник и получател. В зависимост от формата на данните могат да се поддържат различни файлови формати. Например за представяне на изображения могат да се използват файлови формати като JPEG, GIF, TIFF, DIB, BMP и други, за представяне на видео – AVI, MJPEG, MPEG4. Различните операционни системи поддържат различни файлови формати и понякога за да може да се прочете дадена информация, намираща се на друг компютър в мрежата е необходимо преобразуване на информацията. Например ако предаваме през компютърната мрежа филм, записан на диска на нашия компютър, използвайки протокола Digital Living Network Alliance (DLNA) към телевизор, който има мрежов интерфейс, трябва телевизорът да поддържа файловия формат, в който е записан филма. Възможно е обаче да пуснем филма чрез програма, която прави прекодиране – преобразува формата от този, в който файлът е записан към такъв, който телевизорът може да визуализира.

Ако имаме web-страница, написана на кирилица, за да може тя да се прочете правилно, файлът в който е записана (например HTML формат) трябва да бъде съхранен на диска на сървъра във формат UTF-8 (Unicode Transformation Format) или CP1251. Освен това, за правилното визуализиране на страницата, web browser приложението трябва да разбере как да представи страницата, в противен случай вместо букви на кирилица ще бъдат визуализирани странни знаци, например като показаните на фигура 10.2.

За да може да се случи това, обикновено в настройките на Web-сървъра или в скрита част от файла със записаната web страница се поставя информация, указваща формата, на който е написана тя и когато web browser приложението получи и

разпознае успешно тази информация, то визуализира страницата по правилния начин. В противен случай е необходимо потребителя ръчно да променя визуализацията, настройвайки опциите в менюто “Encoding” (кодиране) на web browser-а.




**Фиг. 10.2. Визуализация при неправилен формат на файла.**

### 10.2.2 Компресиране на данни

Компресирането на данни се използва когато искаме да намалим размера на предаваните данни, така че те да заемат по-малко място на диска и да се предават по-малко време през мрежата. Обикновено то разчита на намирането на повтарящи се поредици от битове или символи в оригиналния файл и заместването им с по-къси поредици. Много от съвременните файлове и програми използват компресия – например при снимките форматът JPEG компресира изображението, а форматът BMP го съхранява некомпресирано. При видеофилмите форматът на DVD стандарта – MPEG-2 записва данните некомпресирани, а съвременния стандарт MPEG-4 или неговото развитие H.264 ги компресира, което може да доведе до значителна разлика в размерите на файловете.

При предаванията на данни има два начина за компресия – ако предварително има достъп до всички данни, които ще се предават, целият файл се компресира и тогава се предава по мрежата. Този начин постига по-добра степен на компресия, но изисква предварително познаване и обработка на данните. Ако не е известно какви данни следват обикновено се използва статистическа компресия, която променя своята структура в зависимост от текущо най-често предаваните поредици от битове (символи).

В повечето предавания на данни се изисква да се използва компресия без загуба на информация, при която компресираният файл може да се преобразува обратно до оригиналния некомпресиран. При някои предавания на изображения, видео или аудио сигнал е възможно да се използва компресия със загуба на информация, за да се постигне по-висока степен на компресия, но когато не е необходимо възстановяването на оригиналната информация.

Популярни алгоритми за компресия са ZIP и RAR – формати за компресия на файлове без загуби, MP3 – формат за компресия на аудио със загуби, JPEG (със загуби) и TIFF (с или без загуби) за компресия на изображения, споменатите вече MPEG-4 и H.264 за компресия на видео и други.

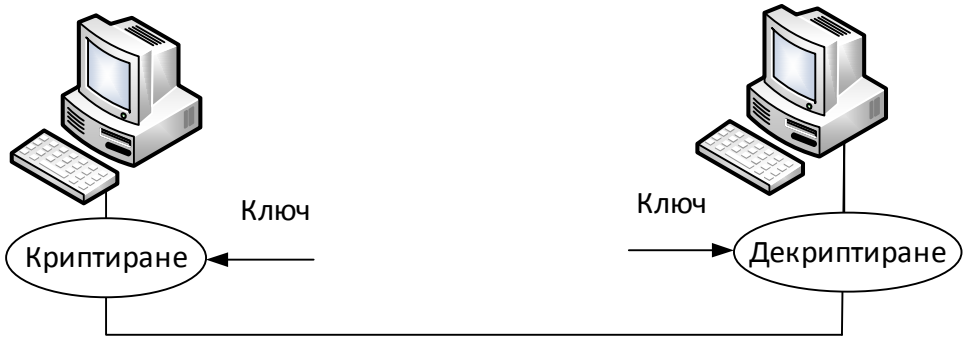
### 10.2.3 Криптиране на данни

Криптирането на данни има за цел да преобразува информацията преди предаване по канала за връзка така, че тя да не може да бъде разчетена от друг, освен от получателя, за който е предназначена. Обикновено криптирането разчита на два компонента – алгоритъм за криптиране, който в повечето случаи е известен и ключ (парола), с която се прави криптирането. Науката за шифриране на съобщения се нарича криптография. Няма алгоритъм, който не може да бъде разбит, целта на всеки алгоритъм за криптиране е да се увеличи времето за разбиване на кода до толкова, че да не си струва влагането на толкова усилия или информацията вече да бъде неизползваема. Някои методи за атаки имат за цел да разкрият криптираното съобщение, а други атакуват самия ключ.

В повечето случаи целта на криптографията е преобразуването на информацията да бъде без загуби – т.е. от криптираната информация да може да бъде възстановена оригиналната. В някои частни случаи целта е да се удостовери изпращаната информация и/или самият изпращач. Тогава криптирането може да използва алгоритми със загуба на информация. Тъй като криптирането променя структурата на предаваната информация, както и компресията на данните, двете функции могат да се използват едновременно – например ако се компресира един файл с използване на парола, той е едновременно компресиран и криптиран.

В съвременния свят се използват няколко вида алгоритми за криптиране на информацията.

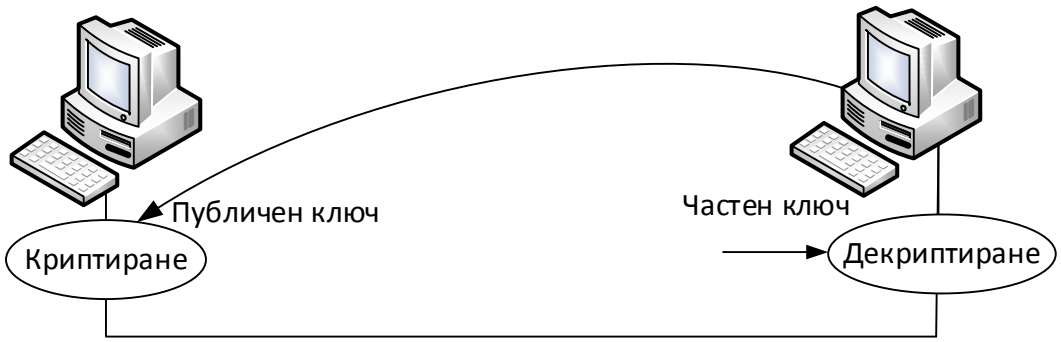
- Симетрични алгоритми, при които паролата за криптиране и декриптиране е еднаква, а алгоритмите са обратни. Примери за такива алгоритми са използваните блокови шифри DES, 3DES и AES, а принципната логика е показана на фиг. 10.3.




**Фиг. 10.3. Симетрична криптография.**

Принципът на работа е следния: източникът и получателят имат предварително избран алгоритъм и ключ. Източникът стартира алгоритъма за криптиране в права посока и криптира информацията, използвайки ключа. Получателят декриптира информацията със същия ключ, използвайки същия алгоритъм в обратна посока. Недостатък на метода е необходимостта от предварителен контакт (обмен на парола) между кореспондентите.

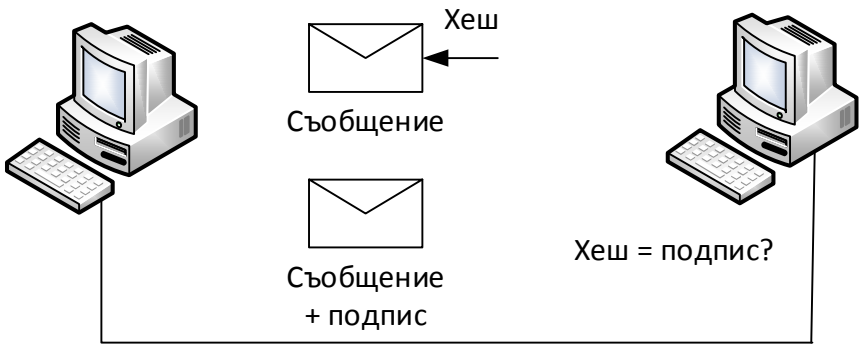
- Несиметрични алгоритми – криптирането и декриптирането се извършва по различен начин, а паролите за криптиране и декриптиране са различни. Най-популярните представители на този клас са алгоритмите с открит ключ (Public-key cryptography), например алгоритъмът RSA, показан схематично на фиг. 10.4.




**Фиг. 10.4. Криптография с открит ключ.**

Сървърът (получателят) изработва двойка ключове – частен и публичен. Те са така подбрани, че с публичния ключ се криптира информацията, а с частния се декриптира. Публичният ключ се предава през мрежата към източника, който криптира информацията с него и я изпраща така по канала. Получателят декриптира информацията с частния ключ. Така не е необходим предварителен контакт между двете страни, тъй като ключа за криптиране се предава по мрежата, но дори някой да го прихване – с него не може да се декриптира предаваната информация.

- Хеш алгоритми – това са алгоритми със загуба на информация, при които не се изисква възстановяването на оригиналната информация. Те най-често се използват за цифрови подписи. Най-често използваните в съвременния свят са MD5 и SHA, като първият вече се счита за компрометиран, но въпреки това е все още широко използван. Принципът на работа е показан на фигура 10.5.




**Фиг. 10.5. Хеш като цифров подпис.**

Преди изпращането на съобщението източникът изработва подписа чрез избраната хеш функция и съдържанието на съобщението. Полученият цифров подпис представлява фиксиран по размер брой битове, който се долепя в края на съобщението. Получателят приема съобщението и подписа, по същата функция от съдържанието на полученото съобщение изработва нов хеш и го сравнява с получения подпис. Ако двете стойности са еднакви, то това е гаранция, че съобщението не е променяно след предаването си. Цифровите подписи не криптират самото съобщение, а само удостоверяват неговото съдържание.

# 11. Приложно ниво

Приложното ниво е единственото с което крайния потребител има пряк контакт – то осигурява мрежовите услуги на приложните програми. Съществуват много разнообразни мрежови приложения. Тук са разгледани само част от често използваните мрежови услуги и приложните протоколи, които ги реализират.

## 11.1 Система за области от имена (Domain Name System, DNS).

Една от най-важните услуги на приложно ниво е системата за области от имена – DNS. Тя позволява на потребителите да работят с имената на сървърите, които са предназначени да бъдат запомняни лесно, например facebook.com и да намират IP адресите, на които са инсталирани тези сървъри.

Системата работи невидимо за крайния потребител, единственото което трябва да направи той е да настрои или да получи автоматично настройки на един или повече DNS сървъри, към които да се изпращат запитванията.

Имената на Интернет сървърите са йерархично организирани. Наи-високото ниво (най-дясната част от името) е областта (домейна) на имената. В началото, когато Интернет е бил разпространен само на територията на Съединените щати са били известни някои области от най-високо ниво (Top Level Domain, TLD), като:

**.gov** – за правителствени организации, например **defense.gov** е министерство на отбраната на Съединените щати;

**.mil** – за военни организации, например **navy.mil** е областта от имена на военноморските сили;

**.com** – за търговски организации, например **ibm.com** е на компанията IBM;

**.net** – за организации поддържащи работата на мрежата Интернет, например регистраторите като RIPE имат сайтове в областта **.net** (**ripe.net**);

**.org** – за неправителствени организации с идеална цел, например първата регистрирана организация е **mitre.org** – некомерсиална компания, управляваща много федерално финансирани изследователски центрове;

**.edu** – за образователни институции, например университетът в Харвард има запазено име **harvard.edu**.

След навлизането на Интернет в другите държави и континенти тези области са оставени за Съединените щати и са избрани различни области за всяка държава, например:

.**de** – за Германия (Deutschland);

.**it** – за Италия;

.**eu** – за Европейския съюз;

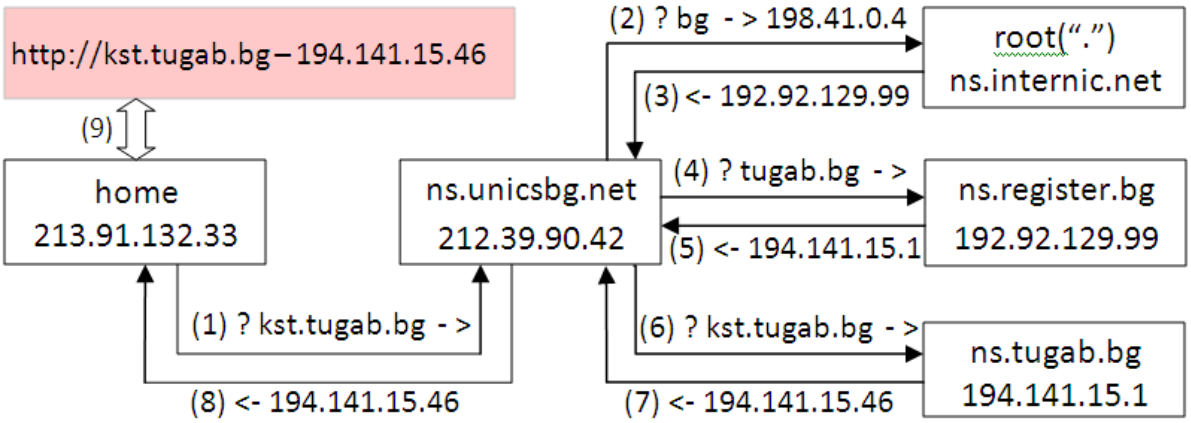
.**bg** – за България и др.

В някои държави се предлага и регистрация в домейни от второ ниво, например във Великобритания за търговски организации често се използва домейнът **.co.uk**.

Въпреки че в някои домейни съществуват рестриктивни правила за регистрация на поддомейни (подобласти), в повечето домейни регистрацията е разрешена за всички и не е проблем персонално име да се регистрира в домейните **.com**, **.org** и др. В България за регистрацията на имена в домейна **.bg** се грижи компанията „Регистър.бг“ и към момента на написване на тази книга таксата за регистрация на име е 30 евро на година. Съществуват доста по-изгодни оферти в големите домейни **.com**, **.net** и др.

Когато даден клиент иска да регистрира име в Интернет, той избира домейна и заявява желаното от него име (ако е свободно), то се нарича поддомейн, например поддомейнът **tugab.bg** е регистриран в домейна **.bg** за технически университет – Габрово. В рамките на този поддомейн могат да се регистрират различни имена за компютри, които дават услуги на Интернет потребителите – например **www.tugab.bg**, **kst.tugab.bg**, **mail.tugab.bg** и други. Тези имена обикновено се администрират локално в организацията, а регистрацията на поддомейна става в т. нар. коренен (root) сървър, който се администрира в организацията, поддържаща домейна.

Така йерархичното име на сървъра **kst.tugab.bg** се регистрира в разпределената база данни DNS по следния начин: домейнът **.bg** е регистриран в корена на дървото – сървъра на Internic (The Internet's Network Information Center) – **ns.internic.net**, поддомейнът **tugab** е регистриран в сървъра на регистър.бг – **ns1.bg**, а името на компютъра **kst** е регистрирано в DNS сървъра на университета – **ns.tugab.bg**. За да отвори даден клиент страницата на сървъра **kst.tugab.bg**, той може да премине през следната поредица от запитвания, показана на фигура 11.1




**Фиг. 11.1. Запитвания при DNS.**

Компютърът на потребителя отваря Интернет браузър и написва в него името **kst.tugab.bg**. За да отвори страницата на него му е необходим IP адреса на сървъра. Тъй като този компютър има настроен DNS сървъра на своя Интернет доставчик, запитването се изпраща към него (в примера 212.39.90.42). Ако той не знае

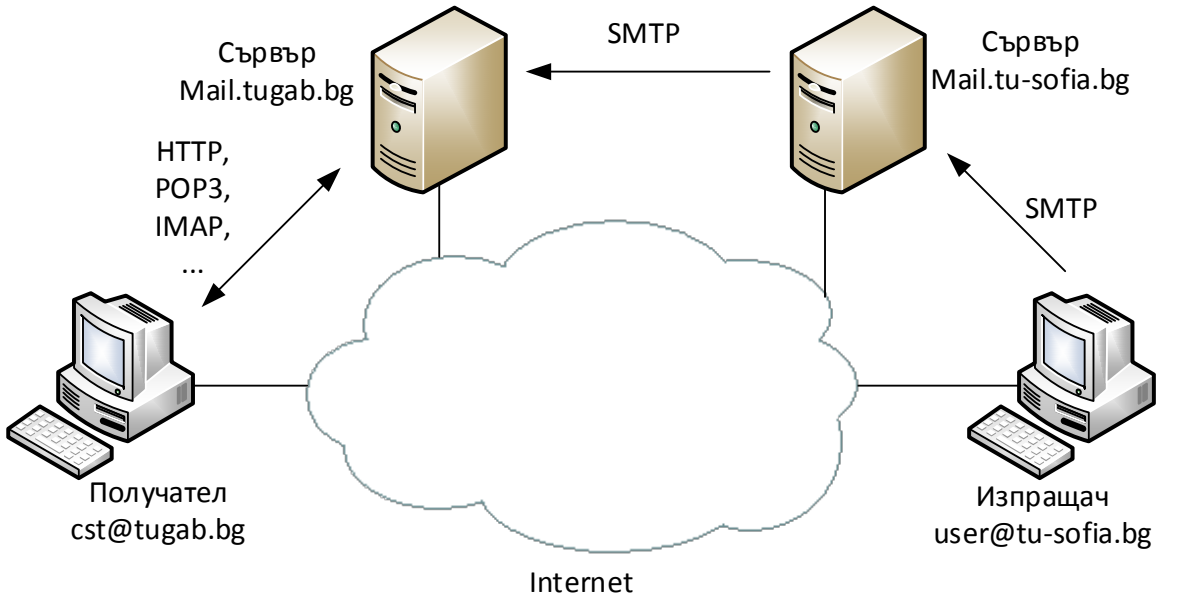
съответствието, изпраща второ запитване към коренния сървър на Internic – на кой адрес се намира сървъра за домейна **.bg**. Той връща IP адреса на сървъра на регистър.бг и към него се изпраща следващо запитване – на кой адрес се намира информацията за поддомейна **tugab.bg**. Този сървър връща като резултат адреса на сървъра на университета и към него се изпраща друго запитване – на кой адрес се намира сървърът **kst.tugab.bg**. След връщането на резултата вече може да се започне установяването на връзка и изтеглянето на страницата на последния сървър.

За да се намали броят на запитванията и времето за изчакване повечето големи организации и Интернет доставчици инсталират свои DNS сървъри, които запомнят известните им съответствия между имена и IP адреси в локален кеш и връщат информацията на потребителя директно от своята памет, вместо да изпращат толкова запитвания. Това обаче означава, че при смяна на IP адреса на даден сайт, клиентите на дадения доставчик няма да научат веднага за промяната, тъй като техния DNS сървър ще продължи известно време да връща стария адрес. Заедно със всеки DNS адрес може да се асоциира определен период от време (TTL – time to live), след изтичането на което даден DNS сървър е длъжен да научи наново съответствието между име и IP адрес.

### 11.2 Електронна поща (e-mail)

Електронната поща е една от най-често използваните услуги в съвременния Интернет свят. Тя дава възможност за обмен на текстови съобщения и прикачени файлове между потребителите, независимо къде се намират и кой Интернет доставчик използват за връзка. Комуникацията не е директна, а преминава през пощенски сървър. Ако двамата потребителя са в една организация могат да използват един и същ пощенски сървър, а ако са различни – комуникацията преминава през два пощенски сървъра – на източника на съобщението и на получателя, както е показано на фигура 11.2.

Изпращачът подготвя електронна поща и я адресира с електронния адрес на получателя – в примера **cst@tugab.bg**. Адресът се състои от две части – име на потребител – **cst** и име на сървър – **tugab.bg**. Използвайки протокола Simple Mail Transfer Protocol (SNMP) изпращачът предава електронната поща на своя пощенски сървър – в примера **mail.tu-sofia.bg**. Пощенският сървър отделя дясната част на адреса, запитва своя DNS сървър за IP адреса на сървъра на получателя – **mail.tugab.bg** и когато го научи, също чрез протокола SMTP предава пощата до сървъра на получателя. Той от своя страна проверява името на потребителя и доставя пощата в съответната пощенска кутия. Протоколът SMTP използва TCP порт номер 25.




**Фиг. 11.2. Комуникация при e-mail.**

Когато потребителят получател реши да провери своята поща, той стартира клиентска програма, чрез която се свързва със своя пощенски сървър. Тази комуникация използва различен протокол. Често използвани протоколи за получаване на електронна поща са Post Office Protocol 3 (POP3), Internet Message Access Protocol (IMAP) и други, но напоследък все по-често пощенските сървъри имат Web интерфейс и комуникацията става чрез протокола за предаване на web страници – HTTP или неговата криптирана версия HTTPS.

## 11.3 Протокол за предаване на хипертекст (HTTP)

Безспорно най-широко използваната Интернет услуга е „Световната паяжина“ (World-Wide Web, WWW), чрез която се предават web страници, намиращи се на web-сървър към клиентите, обикновено с програма web browser (Internet Explorer, Mozila Firefox, Google Chrome, Opera и др.). Почти всички програми и услуги в мрежите и Интернет вече са web-базирани и предлагат възможност за отдалечена работа чрез browser. Основният език за писане на web страници се нарича HyperText Markup Language (HTML), но са популярни и други езици, като PHP, Active Server Pages, Adobe Flash и др. Протоколът, чрез който се предават страниците от сървъра към клиента или се предава информация в обратна посока се нарича протокол за предаване на хипертекст (HyperText Transfer Protocol, HTTP). При него страниците, както и всяка потребителска информация, като имена, пароли и други се предава некриптирана. За да се криптира информацията при предаването между сървъра и клиента се препоръчва използването на сигурната версия на протокола – Secure HTTP (HTTPS).

Комуникацията протича по следния начин: клиентът отваря browser-а и изписва име на сървъра, към който иска да се свърже, например **google.bg**. Компютърът изпраща запитване към DNS сървъра, откъдето научава IP адреса на сървъра,

осъществява TCP съединение с него и започва да изпраща команди чрез HTTP протокола. Най-често използваните команди на протокола са:

- GET – заявка от клиента към сървъра за получаване на някакъв ресурс (напр. Web страница);
- PUT – изпраща ресурси или съдържание към сървъра (напр. файлове или снимки);
- POST – изпраща данни към сървъра, например попълване на форма в сайта.

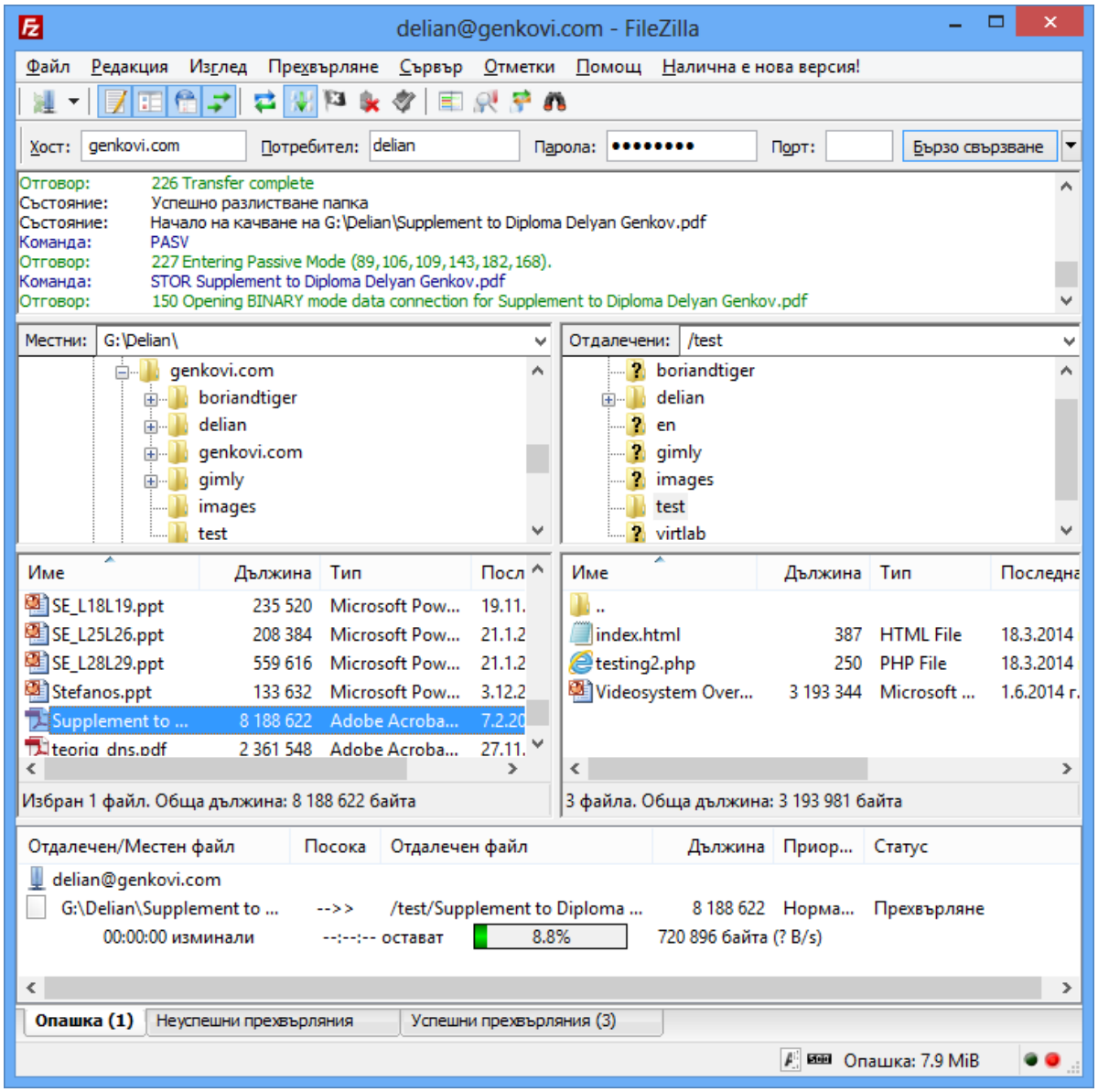
При старите версии на протокола след предаването на страницата до клиента TCP съединението се разпада и при избор на нова хипервръзка от страна на клиента, сочеща към нов ресурс се установява ново съединение. В новата версия има механизми за поддържане на съединението между сървъра и клиента. Също така е възможно с цел ускоряване на предаването някои браузъри да установяват няколко едновременни съединения, например по едно отделно за всеки елемент от страницата – текст, картинка, анимация и др.

### 11.4 Протоколи за предаване на файлове (FTP, TFTP)

Често използван протокол за съхранение на файлове на централен сървър и предаването им към и от потребителите е File Transfer Protocol (FTP). Той също е клиент-сървър базирана услуга. На сървъра се инсталира софтуер и се конфигурират имена на потребители, пароли, права за достъп до файловете и други параметри. Някои популярни FTP сървъри са Filezilla Server за Linux и Windows, Microsoft IIS за Windows и ProFTPD за Linux. Потребителят стартира програма – FTP клиент, с която се свързва към сървъра, при поискване задава потребителско име и парола за достъп, разглежда папките и файловете на сървъра и може да ги изтегля или записва нови файлове на сървъра. Популярни софтуери за FTP клиент са Filezilla Client за Windows и Linux и WinSCP за Windows, въпреки че повечето web browser-и също могат да служат като FTP клиенти.

Един FTP сървър може да бъде настроен да работи в анонимен режим – да не иска потребителско име и парола или да иска стандартното за този режим потребителско име Anonymous и за парола да очаква e-mail адреса на потребителя. Обикновено в този режим достъпът е само за четене. В другия режим потребителят се идентифицира с име и парола и е възможно всеки потребител да има различни права за достъп и да вижда различни файлове, като обикновено потребителят има право да прави всичко със своите файлове – да добавя нови, да ги изтрива или променя и др. Възможно е режимът да е комбиниран – ако се влезе като анонимен потребител файловете да са достъпни само за четене, а при регистриране с име и парола да се получава достъп и за запис.

На фигура 11.3 е показан Filezilla Client с установена връзка към файлов сървър.




**Фиг. 11.3. Filezilla Client установил връзка.**

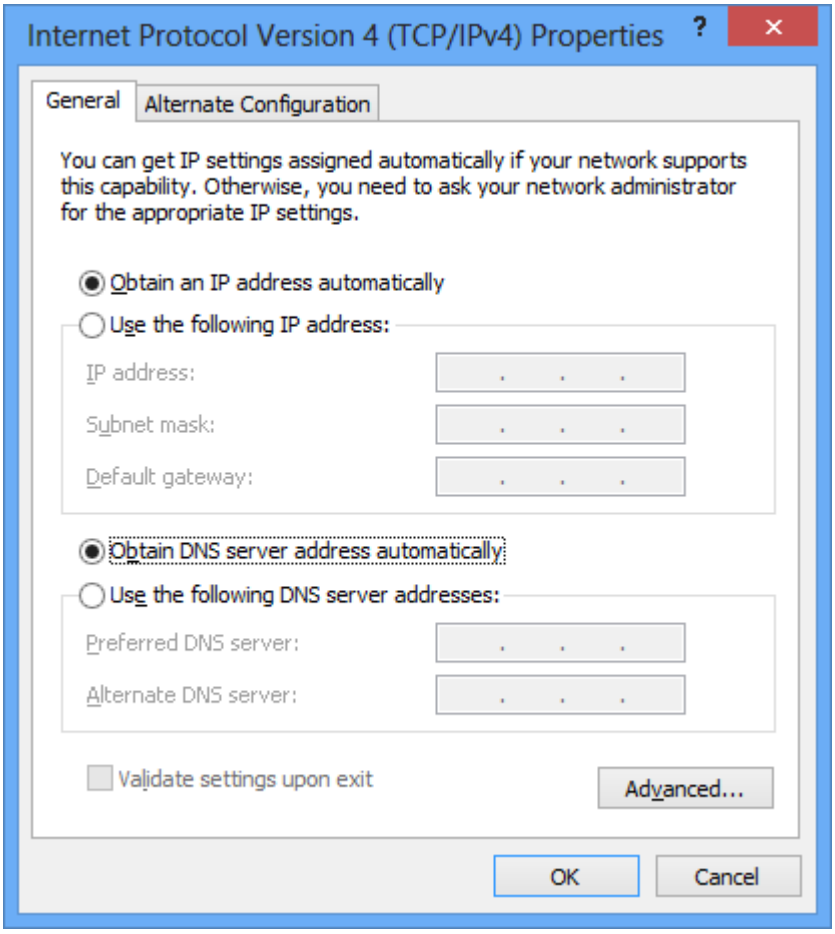
При FTP протокола се отварят едновременно две съединения между клиента и сървъра – едно за команди на TCP порт 21 и второ за данни. Класическите FTP сървъри работят на TCP порт за данни 20, но могат да бъдат настроени в пасивен режим (PASV), при който номерът или номерата на портовете за данни да се договарят динамично през порта за команди и тогава обикновено те са с номера над 1024. Протоколът FTP предава файловете, командите и данните (потребителско име, парола) некриптирани. Съществува версия SFTP, която криптира комуникацията.

Опростената версия на протокола се нарича Trivial File Transfer Protocol (TFTP). При нея няма потребителски имена и пароли и трансферът се извършва с транспортен протокол UDP на порт 69. Този протокол често се използва за по-лесно предаване на файлове в локални мрежи между сървър и мрежови устройства – маршрутизатори,

комутатори. При Windows обикновено е необходимо използването на външна програма, популярни TFTP програми са Tftpd32 и SolarWinds TFTP Server.

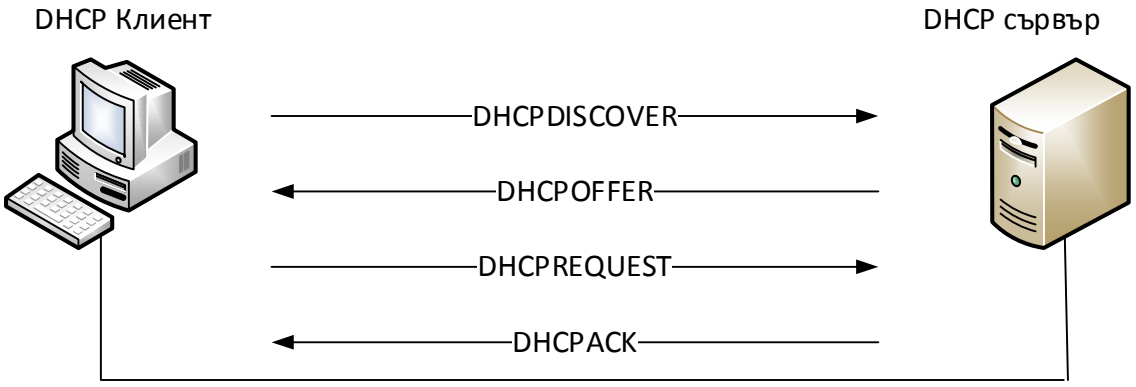
## 11.5 Автоматично назначаване на параметри (DHCP)

Протоколът за автоматично назначаване на IP параметри (Dynamic Host Configuration Protocol, DHCP) е доста удобен, когато се използва мобилно устройство, което се включва в различни мрежи с различни IP адресни схеми. Вместо да се конфигурира всеки път ръчно адрес, мрежова маска, шлюз по подразбиране и адреси на DNS сървъри, може да се остави устройството на автоматично получаване на адресни параметри, както е показано на фигура 11.4 за Windows 8.




**Фиг. 11.4. Автоматични настройки за IP параметри.**

В този случай при зареждане на операционната система или стартиране на мрежовата връзка клиентът чрез протокола DHCP изпраща пакет, наречен DHCPDISCOVER до broadcast IP адрес 255.255.255.255, търсейки услугите на DHCP сървър в мрежата, който да му назначи IP адресни параметри. При наличие на такъв сървър диалогът протича по показания на фигура 11.5 начин.




**Фиг. 11.5. Диалог при DHCP.**

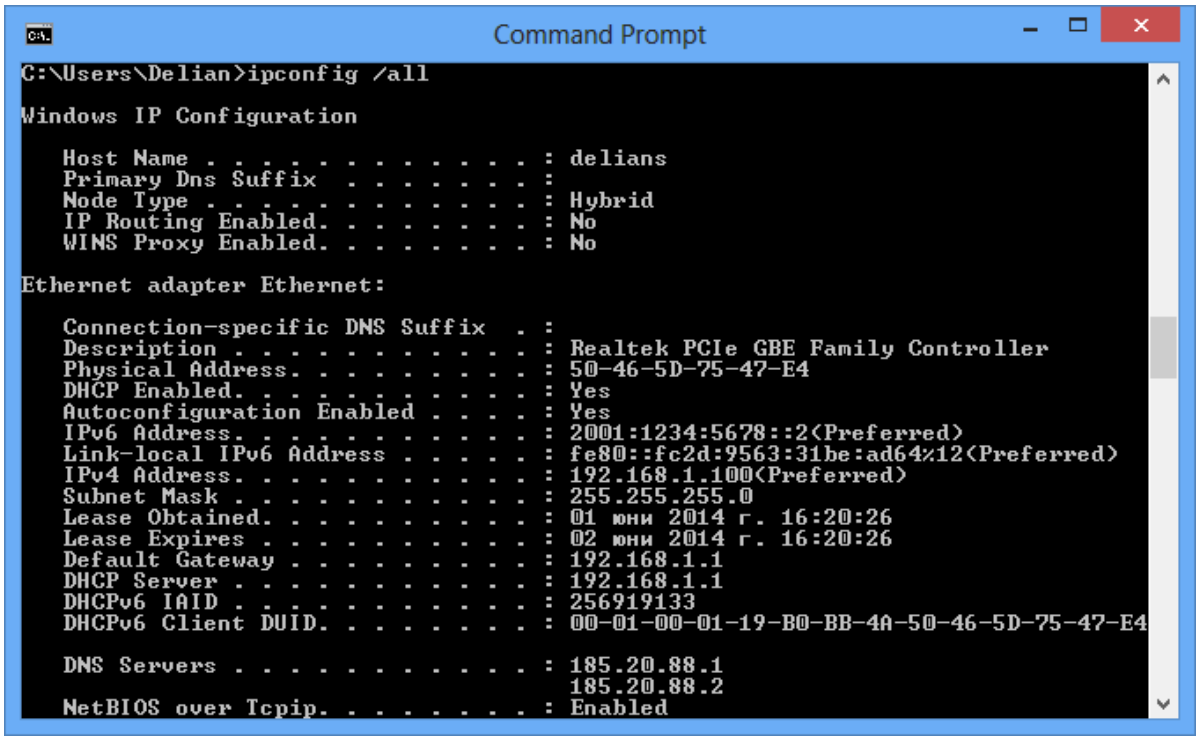
Сървърът връща пакет DHCPOFFER, съдържащ параметрите, предлагани на клиента – IP адрес, мрежова маска и др. В повечето реализации този пакет се изпраща само до MAC адреса на получателя. Третият пакет DHCPREQUEST е заявка от клиента към сървъра, че желае да използва предложените му IP адресни параметри. Този пакет също се изпраща като broadcast. С последния пакет DHCPACK сървърът потвърждава назначаването на адреса на клиента. Съобщенията на протокола се изпращат на UDP порт 67 за сървъра и UDP порт 68 към клиента.

Един от параметрите на DHCP сървъра е времето за назначаване на адреса – Lease time. Той може да е няколко минути, няколко часа, няколко дни или безкрайно. Когато този параметър е настроен на сървъра, клиентът е длъжен на 50% от времето за назначаване на адреса да се опита да поднови адреса си за още един период.

Тъй като за някои устройства (сървъри, принтери) е важно винаги да имат едни и същи адреси, те могат или да се адресират статично, или в DHCP сървъра да бъде направена резервация – да се определи, че на този MAC адрес винаги ще се назначава даден IP адрес.

Протоколът DHCP обикновено се използва за назначаване на IP адрес, мрежова маска, шлюз по подразбиране и адреси на DNS сървъри, но съществуват и други параметри, които могат да бъдат назначавани, както и доста производители са направили свои разширения на протокола. По този начин чрез DHCP могат да се назначават голям брой допълнителни параметри. Например при IP телефоните на Cisco Systems важен параметър за настройка е „опция 150“, която позволява на IP телефоните да се назначава адрес на TFTP сървър, от който те да си изтеглят конфигурационни файлове.

Параметрите на услугата DHCP под Windows клиент могат да бъдат преглеждани или манипулирани с помощта на командата ipconfig, както е показано на фигура 11.6.




**Фиг. 11.6. Преглед на DHCP параметри с ipconfig /all.**

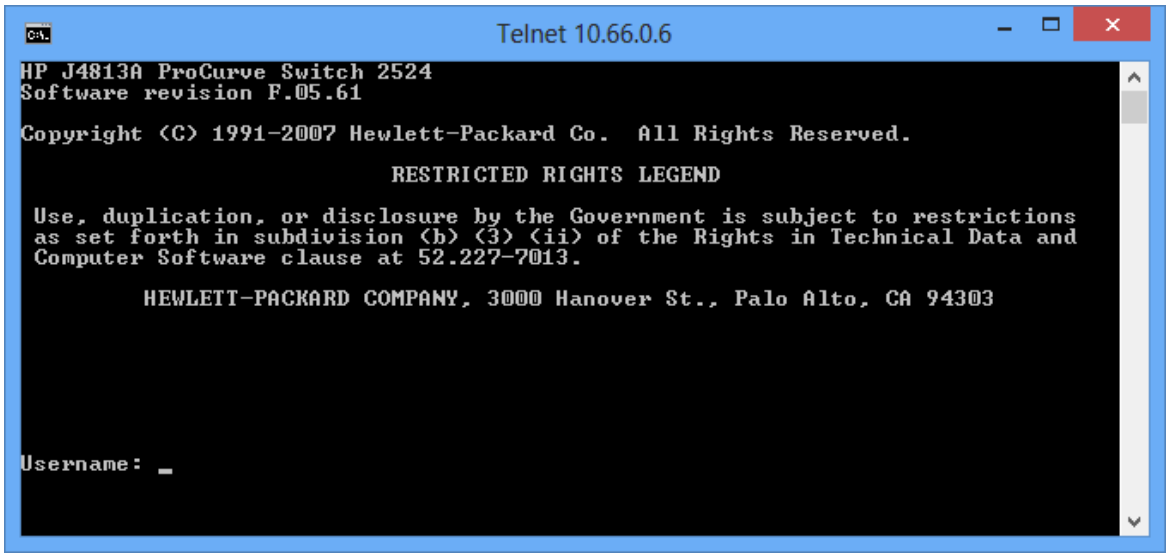
Чрез командата **ipconfig /all** се виждат параметрите, които са получени, адресът на DHCP сървъра, началото и края на времето за наемане на адресните параметри (lease time). С командата **ipconfig /release** може да бъде освободен получения IP адрес, а с **ipconfig /renew** – да се изпрати повторно искане за назначаване на адрес.

## 11.6 Отдалечен достъп (Remote access)

Отдалеченият достъп до устройства и компютри в мрежата е важен, тъй като голяма част от сървърите и мрежовите устройства стоят в специални сървърни помещения, разположени в компютърни шкафове и рядко към тях има свързани монитори, клавиатури и мишки за да се наблюдават и управляват техните състояния и параметри. Някои от тях имат специализирани web сървъри и отдалечения достъп се прави чрез web браузър и HTTP протокол. За други се използват специални протоколи за отдалечен достъп.

### 11.6.1 Telnet

Протоколът telnet е текстово базиран отдалечен достъп до устройства – маршрутизатори, комутатори или сървъри. Той предполага устройството да има вграден telnet сървър, който по подразбиране работи на TCP порт 23, а на компютъра от който осъществяваме отдалечения достъп да се стартира telnet клиент. В командния ред на Windows и linux има telnet клиенти, като за Windows 7 и 8 той трябва да бъде инсталиран допълнително. Примерен достъп към комутатор чрез telnet е показан на фигура 11.7.

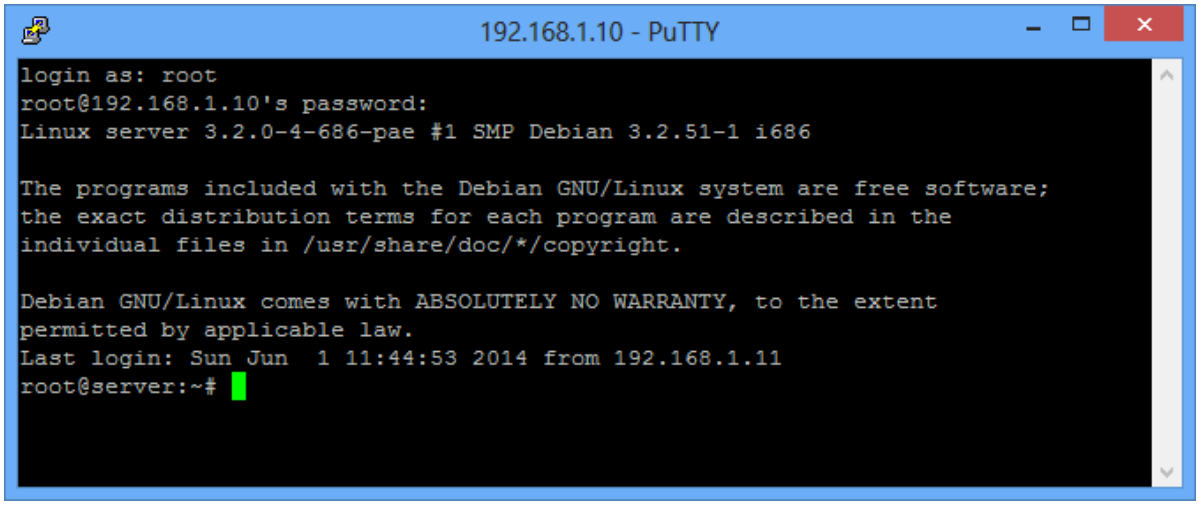



**Фиг. 11.7. Telnet достъп до комутатор.**

При Telnet обменът на данни не е криптиран и командите, имената и паролите се предават в открит вид по мрежата, следователно могат да бъдат подслушани.

### 11.6.2 Secure Shell (SSH)

Протоколът SSH е криптираната алтернатива на telnet – текстово базиран протокол с криптиране на връзката. Затова той често се предпочита и е подразбиращия се начин за отдалечен достъп до повечето съвременни linux сървъри. Във Windows няма вграден SSH клиент, затова се ползва външна програма, например показаната на фигура 11.8.




**Фиг. 11.8. SSH достъп до Linux сървър.**

Протоколът SSH по подразбиране използва TCP порт 22 за достъп до сървъра.

### 11.6.3 Remote Desktop Protocol (RDP)

За графичен отдалечен достъп често се използва вградената във Windows и в някои други системи Remote Desktop. Тя работи по подразбиране на TCP порт 3389 и дава възможност за работа с графичната конзола на отдалечения компютър. Пример за отдалечено свързване към Windows сървър е показан на фигура 11.9.


**Фиг. 11.9. Remote Desktop достъп до Windows сървър.**

Обикновено осъществявайки отдалечен достъп към Windows чрез Remote desktop локалният потребител се изключва от компютъра и не може да работи докато отдалечения потребител е свързан към системата. При някои сървърни системи с добавен лиценз може да работят няколко потребителя едновременно на една машина, като всеки от тях вижда свой екран.

### 11.6.4 Virtual Network Computing (VNC)

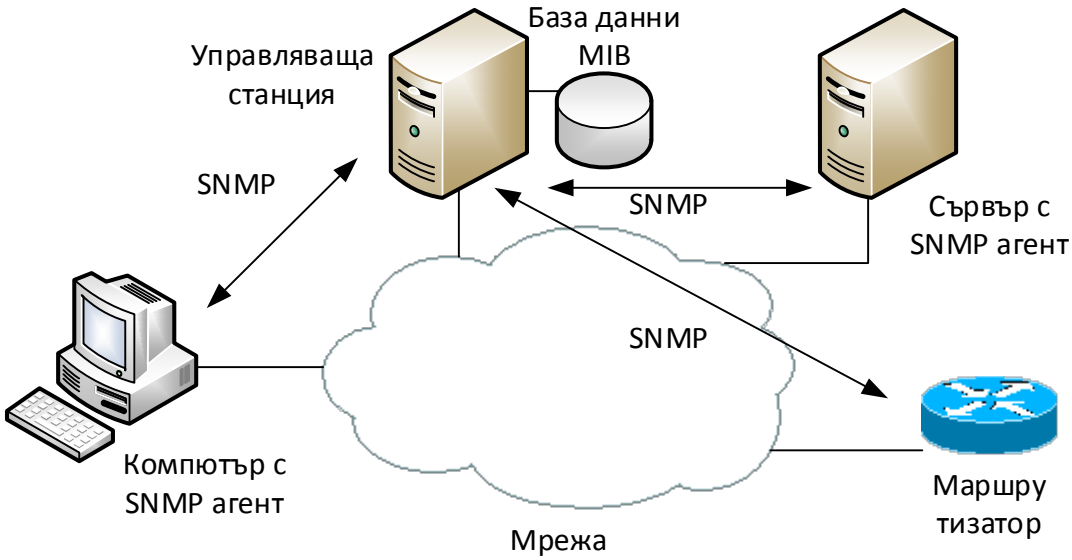
Алтернатива на графична система за отдалечен достъп с отворен код, достъпна за Windows и linux е VNC, която има доста различни реализации – RealVNC, TightVNC, UltraVNC и други, част от които безплатни, други платени. По подразбиране работи на TCP порт номер 5900, някои системи предлагат възможност за конфигуриране на виртуални дисплей за различните потребители, при което номерът на порта е 5900 + номера на виртуалния дисплей. При свързване чрез VNC локалният потребител не се изключва, а двамата виждат един и същ екран, дори при определени настройки могат

едновременно да командват клавиатурата и мишката, което го прави удобен за демонстрационни цели или отдалечена помощ.

Освен описаните в тази глава програми за отдалечен достъп съществуват и много други, като Team Viewer, DameWare, LogMeIn, всяка от които си има особености и различна лицензионна политика.

## 11.7 Мрежово наблюдение и управление (SNMP)

За централизирано наблюдение на параметри на мрежови устройства и компютри, както и за тяхното управление е предвиден протоколът Simple Network Management Protocol (SNMP). Необходимите компоненти за него са показани на фигура 11.10.




**Фиг. 11.10. Структура на SNMP наблюдение.**

В типичните си приложения SNMP използва един или няколко компютъра с инсталиран специализиран софтуер, наречен управляваща станция или мениджър, който наблюдава или управлява група мрежови устройства. На устройствата работи друг специализиран софтуер, наречен агент, който отговаря на запитвания от мениджъра.

Агентите организират данните за наблюдение или управление в специални променливи. Променливите са организирани йерархично и тяхната организация, типове и описания са събрани в Management Information Base (MIB) – база данни за управление на устройството.

Дефинирани са седем типа съобщения:

- GetRequest: заявка от мениджъра към агента за получаване на стойност на променлива или списък стойности.
- SetRequest: заявка от мениджъра към агента за промяна на стойност на променлива или списък стойности.

- GetNextRequest: заявка от мениджъра към агента за откриване на достъпните променливи и стойностите им.
- GetBulkRequest: оптимизирана версия на GetNextRequest за много итерации.
- Response: отговор от агента, връщащ стойност или потвърждаващ заявка.
- Trap: асинхронно уведомяване от агента към мениджъра за настъпило събитие.
- InformRequest: потвърждение на Trap от мениджъра към агента.

Обикновено управляващата станция обхожда всички устройства, предавайки им Get заявки, на които те отговарят със текущите стойности на своите параметри – трафик на мрежовите интерфейси, свободно място на твърдия диск, натоварване на процесора, заета оперативна памет и много други. Получените параметри се записват в базата данни и могат да се обработват – да се чертаят графики, да се анализират, да се реагира на настъпили събития и др. Ако системата се използва и за управление, при настъпване на дадено събитие е възможно да се изпрати Set заявка към устройството, принуждаваща го да извърши дадено действие. Ако е конфигурирано е възможно устройството да изпрати Trap - информация за възникване на дадено събитие, без да чака запитване от упралвяващата станция.

Йерархичната база данни (MIB) може да съдържа стандартна за всички устройства информация, както и специфична за даден производител и/или устройство. Информацията се състои от множество идентификатори на обекти (Object identifiers, OID), всеки от които описва един параметър или група параметри на устройството. Всеки идентификатор е поредица от числа и/или думи, разделени с точки, например за да се получи името, назначено на дадена мрежова система, трябва да се прочете идентификаторът 1.3.6.1.2.1.1.5.0, като поредицата числа описва пътя от корена на дървото на базата данни до конкретната стойност.

Връзката между мениджъра и агента се осъществява чрез въвеждане на Community string – дума, играеща роля на парола. Обикновено на устройствата могат да се задават две нива на пароли – RO, която се използва само за четене на параметрите и RW, която може да се използва и за запис на стойности. За реализация на Set заявки е необходимо да се разполага със стринга за четене и запис.

В момента съществуват три различни версии на протокола – версия 1, версия две с подверсии 2c и 2u и версия 3. Версия 3 поддържа криптиране на съобщенията и автентикация, което я прави доста по-сигурна от останалите, но още не всички устройства я поддържат.

## Списък на използваните термини и съкращения

**ARPANET** – първата мрежа с протоколи TCP/IP, в последствие гръбнак на Интернет.

**Access Point** – точка за достъп, вид безжично мрежово устройство.

**Accounting (отчитане)** - отчитане на използваните ресурси.

**Ad-hoc** – двуточково безжично предаване на данни.

**Anycast** – метод за предаване на данни при IPv6 до един от дадена група получатели.

**APIPA (Automatic Private IP Addressing)** – адресиране с автоматични локални адреси.

**ARP (Address Resolution Protocol)** - протокол за съвпадения на адреси, намиращ MAC адрес по познат IP адрес.

**AS (Autonomous Systems)** - автономни системи, начин за разделяне на Интернет.

**Authentication (автентикация)** – проверка за идентичност на потребителя.

**Authorization (оторизация)** – проверка на правата на потребителя до мрежови ресурси.

**Auto MDI/MDI-X** – функция за автоматична размяна на приемна и предавателна двойка

**BGP (Border Gateway Protocol)** – външният маршрутизиращ протокол, използван в Интернет.

**bit (бит)** - основна единица информация в компютрите, представляваща най-малката информация, която може да се съхранява или предава. Може да има стойност 0 или 1.

**Broadcast (броудкаст)** – метод за предаване до всички в дадена мрежа.

**Browser, web browser** – програма за разглеждане на web страници.

**Byte (байт)** - съвкупност от осем бита. Единица информация, която може да представи една буква в писмо или документ.

**bps (бит в секунда)** - мярка за скорост на предаване на информация, при която за една секунда се предава един бит.

**Bps (байт в секунда)** - мярка за скорост на предаване на информация, при която за една секунда се предава един байт.

**BNC (British Naval Connector)** – съединител за коаксиален кабел.

**Checksum (Контролна сума)** - метод за проверка за грешки.

**CRC (Cyclic Redundancy Check)** - метод за проверка за грешки.

**CSMA/CD (Carrier Sense Multiple Access with Collision Detection)** – дисциплина на достъп в Ethernet мрежи.

**Default Gateway** – шлюз по подразбиране, конфигурационен параметър на протокол IP.

**Default Route** - път по подразбиране, специален адрес за маршрутизаторите.

**DHCP (Dynamic Host Configuration Protocol) –** протокол за автоматично конфигуриране на параметри.

**DMZ** - демилитаризирана зона, по-малко защитена зона от вътрешната мрежа.

**DNS (Domain Name System)** - Система за именуване на области, начин за намиране на адрес по име.

**DoD (Department of Defense)** - Министерството на отбраната на Съединените щати.

**EGP (Exterior Gateway Protocols) –** външни маршрутизиращи протоколи.

**EIA/TIA 568A, EIA/TIA 568B –** стандарти за поставяне на конектор на кабел усукана двойка проводници.

**Encoding (кодиране) –** представяне на информация в даден формат.

**EIGRP (Enhanced Interior Gateway Routing Protocol) –** маршрутизиращ протокол за Cisco Systems, вече отворен стандарт.

**Ethernet (Етернет) –** вид компютърна мрежа, най-често използвана в момента за изграждане на локални мрежи.

**e-mail –** електронна поща.

**EUI-64 –** техника за автоматично изчисляване на дясната част на IPv6 адрес.

**FDDI (Fiber Distributed Data Interface) –** стар вид оптични компютърни мрежи с кръгова топология.

**Frame (кадър) –** единица данни на канално ниво.

**FTP (Foiled Twisted Pair) –** кабел екранирана с фолио усукана двойка проводници.

**FTP (File Transfer Protocol) –** протокол за предаване на файлове през мрежа.

**Header –** заглавна част на протокол.

**Host –** компютър или крайно устройство в мрежата.

**HTML (HyperText Markup Language) –** език за създаване на web страници.

**HTTP (HyperText Transfer Protocol)** - протокол за предаване на web страници.

**IANA (Internet Authority Numbers Association**) – организация, разпределяща номерата в Интернет.

**ICMP (Internet Control Message Protocol)** - протокол за управляващи съобщения в Интернет.

**IEEE (Institute of Electrical and Electronics Engineers) –** институт на инженерите по електротехника и електроника.

**IETF (Internet Engineering Task Force) –** организация, грижеща се за поддръжката и развитието на Интернет.

**IGP (Interior Gateway Protocols)** – вътрешни маршрутизиращи протоколи.

**IMAP (Internet Message Access Protocol)** - протокол за приемане на електронна поща.

**Internet (Интернет)** - световната компютърна мрежа, свързваща милиарди компютри в целия свят.

**IP (Internet Protocol)** – протокол за комуникация в Интернет.

**IPv4** – текущата четвърта версия на Интернет протокола.

**IPv6** – бъдещата шеста версия на Интернет протокола.

**ISO (international Standarts Organization) –** международна стандартизационна организация

**ISP (Internet Service Providers)** – доставчици на Интернет услуги.

**Gbps (гигабит в секунда)** - мярка за скорост на предаване на информация, при която за една секунда се предават 1024 мегабита или 1 073 741 824 бита.

**kbps (килобит в секунда)** - мярка за скорост на предаване на информация, при която за една секунда се предават 1024 бита.

**kBps (килобайт в секунда)** - мярка за скорост на предаване на информация, при която за една секунда се предават 1024 байта.

**LAN (Local Area Network, локална мрежа)** - мрежа, свързваща компютърните системи, намиращи се в една стая, етаж сграда или в група сгради на близко разстояние.

**LLC (Logical Link Control)** - логическо управление на връзката, част от каналното ниво.

**Local Host (Loopback)** – специален адрес, означаващ локалния компютър.

**MAC (Media Access Control)** – управление на достъпа до средата – дисциплина на компютърната мрежа, част от каналното ниво.

**MAC address** – метод за адресиране в многоточкови мрежи.

**Mbps (мегабит в секунда)** - мярка за скорост на предаване на информация, при която за една секунда се предават 1024 килобита или 1048576 бита.

**MBps (мегабайт в секунда)** - мярка за скорост на предаване на информация, при която за една секунда се предават 1024 килобайта или 1048576 байта.

**Management Information Base (MIB)** – база данни за параметри на устройства, използвана от протокол SNMP.

**MIMO (Multiple In Multiple Out)** – технология за безжично предаване на данни с по-висока скорост.

**MPLS (Multi Protocol Label Switching)** – съвременен метод за изграждане на мрежи на доставчици на услуги.

**MTU (Maximal Transmission Unit)** - максимален размер на пакета.

**Multicast** – вид предаване на данни до няколко получателя едновременно, многоцелеви.

**Multi-Mode** - вид оптично влакно.

**NAT (Network Address Translation)** - превод на мрежови адреси, техника за достъп до Интернет чрез частни адреси.

**NRZ (Non-return to Zero)** – вид кодове за представяне на сигнал без връщане в нулата.

**OSI (Open System Interconnection)** – препоръчителен модел за взаимодействие между отворени системи.

**OSPF (Open Shortest Path First)** – сложен маршрутизиращ протокол, подходящ за големи мрежи.

**OUI (Organization Unique Identifier)** - уникален идентификатор на организацията, лявата част от MAC адреса.

**Packet (пакет)** – единица данни на мрежово ниво.

**PAN (Personal Area Network, персонална мрежа)** - компютърна мрежа свързваща личните устройства на един човек – смартфон, таблет, лаптоп, персонален компютър.

**Parity check (контрол по четност)** – метод за проверка за грешки.

**PAT (Port Address Translation)** - техника за достъп до Интернет на много частни адреси през по-малко публични.

**peer-to-peer (равноправни мрежи)** - мрежи в които всеки компютър може да дава услуги на останалите в мрежата, както и да използва услуги от останалите.

**PMTUD (Path MTU Discovery)** - алгоритъм за избягване на фрагментацията.

**POP3 (Post Office Protocol 3)** – протокол за приемане на електронна поща.

**POS (Point-of-Sale)** – устройство за плащане с карта.

**QoS (Quality of Service)** - качество на обслужването, механизъм за приоритизация.

**RDP (Remote Desktop Protocol)** – протокол за графичен отдалечен достъп.

**RIP (Routing Information Protocol)** – прост маршрутизиращ протокол, избиращ най-късия път.

**RJ-45** – съединител за кабел усукана двойка проводници.

**RG-59** – вид коаксиален кабел, наричан още „тънък“.

**RG-8** - вид коаксиален кабел, наричан още „дебел“.

**Router (маршрутизатор)** – устройство за свързване към глобална мрежа или Интернет.

**Routing (маршрутизация)** – процес на изор на най-добър път за пакет.

**RZ (Return to Zero)** - вид кодове за представяне на сигнал с връщане в нулата.

**Segment (сегмент)** – единица данни на транспортно ниво.

**SFTP (Shielded Foiled Twisted Pair)** - кабел екранирана с оплетка усукана двойка проводници.

**SFTP (Secure File Transfer Protocol)** – сигурен протокол за предаване на файлове.

**Single Mode** – вид оптично влакно.

**SMTP (Simple Mail Transfer Protocol)** – протокол за предаване на електронна поща.

**SNMP (Simple Network Management Protocol)** – протокол за мМрежово наблюдение и управление.

**SOHO (Small-office Home-office)** - малка локална мрежа, свързваща няколко компютърни системи в дома или в малък офис.

**SSH (Secure Shell)** – криптиран протокол за отдалечен достъп.

**Switch (комутатор)** – устройство за свързване на компютри в локална мрежа.

**TCP (Transmission Control Protocol)** – протокол за управление на предаването, част от Интернет.

**TCP/IP** – основните протоколи на Интернет.

**TFTP (Trivial File Transfer Protocol)** – прост протокол за предаване на файлове.

**TLD (Top Level Domain)** - област от най-високо ниво, домейн, крайна част на Интернет името.

**TOS (Type of Service)** - тип на услугата, метод за приоритизиране на пакети.

**TTL (Transistor–transistor logic)** – цифрова логика, при която единицата се представя с напрежение +5V, а нулата с напрежение 0V.

**TTL (Time To Live)** - време на живот, параметър на IP протокола, указващ максималният престой на пакета в мрежата.

**UDP (User Datagram Protocol)** – протокол за потребителски дейтаграми, част от Интернет.

**UTF-8 (Unicode Transformation Format) –** универсален формат за документи, поджодящ за кирилица и други азбуки.

**UTP (Unshielded Twisted Pair)** – кабел неекранирана усукана двойка проводници.

**VLSM (Variable Length Subnet Mask)** – Маска на мрежата с променлива дължина, техника за разделяне на мрежа на различни по размер подмрежи.

**VNC (Virtual Network Computing)** – протокол за графичен отдалечен достъп.

**VPN (Virtual Private Networks)** - виртуални частни мрежи.

**Token Ring** – вид компютърна мрежа на IBM с кръгова топология.

**WAN (Wide Area Network, глобална мрежа)** - мрежа, покриваща няколко съседни града, цяла страна, няколко страни, цял континент или дори няколко континента.

**WEP (Wired Equivalent Privacy)** – стар метод за защита на безжични мрежи.

**Wi-Fi (Wireless Fidelity)** – група технологии за безжично предаване на данни.

**WPA2 (Wi-Fi Protected Access)** – съвременен метод за защита на безжични мрежи.

**WWW (World-Wide Web)** - „Световната паяжина“, услугата от свързани web страници.

## Използвана литература

1. Cotton, M, L. Eggert и др., „Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry“, IETF, RFC 6335, August 2011.
2. Cisco Systems, “Internetworking Technologies Handbook (2nd Edition)”, ISBN: 1-56205-102-8
3. Cisco Systems, “Cisco Networking Academy Program, Semesters 1 – 4”, Online course program, 2012-2014
4. Marine, A., J. Reynolds, G. Malkin, „FYI on Questions and Answers - Answers to Commonly asked New Internet User Questions“, IETF, RFC 1594, March 1994.