14. Компютърни мрежи: Компютърни мрежи и протоколи – OSI модел. Канално ниво. Маршрутизация. IP, TCP, HTTP.

OSI модел – най-обща характеристика на нивата. Канално ниво – кадри, прозорци, предаване и грешки. Какво е характерно за Ethernet. Статична маршрутизация – таблица и избори. Централизирана маршрутизация – недостатъци. Разпределена маршрутизация – алгоритъм с дистантен вектор, алгоритъм със следене състоянието на връзката. IP дейтаграма. IP адресация – класова, безкласова, преобразуване на IP в MAC и обратно. TCP – 3-way hand shake. Формат. Разлика с UDP. HTTP.

OSI модел – най-обща характеристика на нивата.

Стандартът **OSI** (Open System Interconnection), разработен от Международната организация за стандартизация (International Standard Organization – ISO) стандартизира начините за връзка между отворени системи. В случая терминът „отворена система" означава система, чиито ресурси могат да се използват и от другите системи, образуващи мрежата. Важен принцип е разслояването, което разделя логически всяка от системите, образуващи мрежата, на йерархично подредени подсистеми. Слоевете са 7 и се номерират отдолу нагоре. Всеки слой осигурява определено обслужване за по-горния, като използва функциите на по-долния.

7	Приложен слой
6	Представителен слой
5	Сесиен слой
4	Транспортен слой
3	Мрежов слой
2	Канален слой
1	Физически слой

- **Физически слой** (*Physical layer*): Обектите от този слой са технически средства, реализиращи предаването на данни през определена физическа среда, т.е. това е нивото на физическите връзки. Основната му функция е да управлява кодирането и декодирането на сигналите, интерпретиращи двоичните цифри 0 и 1, без да се интересува от предназначението на тези битове и на предаваните данни. В този слой предавателят има за цел да изпрати поредица от кодирани 0 и 1, а приемникът да ги декодира в цифров вид.

- **Канален слой** (*Data-link layer*): Този слой реализира връзките на логическо ниво, т.е. се занимава с обмена на данни, без да го интересува начинът, по който те се преобразуват в електрически сигнали във физическия слой. Типична функция е откриването и коригирането на грешки при предаването на данните. Обикновено данните на канално ниво се обменят на порции с фиксирана дължина, наречени кадри, чийто формат се определя от избрания протокол за предаване на канално ниво. Функциите на този слой обикновено се реализират смесено – апаратно и програмно.

- **Мрежов слой** (*Network layer*): Този слой реализира връзките на мрежово ниво. Занимава се с изпращането и получаването на пакети (дейтаграми) и доставката им (маршрутизация). Този слой отговаря и за адресацията на машините в мрежата. Комутирането на пакетите е друга важна задача, както и предотвратяването на претоварвания в мрежата. Функциите на мрежовия, както и на всички по-горни слоеве, обикновено се реализират програмно.

- **Транспортен слой** (*Transport layer*): Осигурява транспортирането на съобщения от системата-източник до системата-приемник и представлява надстройка над мрежовия слой. Слоят управлява обмена на данните и осъществява връзка от тип „край-край". Гарантира за надеждното и ефективното транспортиране на данните. Неговите действия остават скрити за по-горните слоеве по отношение на управлението и движението на

потока от данни от единия хост към другия. Функциите, изпълнявани от транспортния слой, отговарят за цялостта на управлението на диалога между крайните потребители, включително откриване на загубени съобщения и повторното им изпращане.

- **Сесиен слой** (*Session Layer*): Отговорен за диалога между две комуникиращи програми и за управление на обмена на данни между тях, използва интерактивен диалог между двете програми. Дефинира 3 типа диалози: двупосочен едновременен, двупосочен алтернативен и еднопосочен диалог. При организиране на диалога в сесийния слой се включват синхронизиращи елементи, които позволяват прекъсване на диалога и в последствие възстановяването му от мястото на прекъсването.

- **Представителен слой** (*Presentation Layer*): Най-ниският слой, който се интересува от значението на потоците обменяни битове. Грижи се за запазването на информационното съдържание на данните. Има две основни функции: да договаря общ синтаксис за предаване на съобщенията и да осигурява възможност едната програмна система да не се грижи какъв е форматът на вътрешната структура на представените данни, които другата система използва. Ако двете системи са например персонални компютри, работещи на Basic, представителният слой няма особени функции, тъй като двете програми имат общи вътрешни структури на данните. В противен случай обаче представителният слой извършва необходимото преобразуване, като позволява на всяка програма да работи със собствения си формат, без да се интересува от формата, който използва другата програма.

- **Приложен слой** (*Application Layer*): Най-горният слой, към който се свързват потребителските процеси. Предоставя средства за работа с приложни програми, за които осигурява достъп до системите за осъществяване на комуникациите. На ниво приложен слой програма, работеща на един хост, изпраща съобщение, което се получава от програма, работеща на друг хост, и този слой не се интересува как точно съобщението на програмата от предаващия компютър е стигнало до приемащия.

При изпращане на данни всеки слой възприема цялата информация, получена от по-горния слой като данни, и добавя своя собствена управляваща информация, наречена заглавна част (header), за да осигури правилна доставка на информацията. Добавянето на информация за осигуряване на доставката се нарича опаковане. При получаване на данни от по-долен слой настъпват процеси, обратни на горните. Всеки слой премахва своята заглавна част, преди да подаде данните на по-горния слой. Информацията, получена от по-долен слой, се интерпретира като съвкупност от заглавна част и данни.

Канално ниво – кадри, прозорци, предаване и грешки.

Каналното ниво е второто ниво в седемслойната архитектура, между физическия и мрежовия слой. То организира прехвърлянето на данни от мрежовия слой на машината-източник към мрежовия слой на машината-получател. Обектите на този слой реализират връзките на логическо ниво, т.е. не се интересуват от начина, по който данните се преобразуват в електрически сигнали от съоръженията, работещи на физическия слой. Функциите на канално ниво се реализират смесено – апаратно и програмно.

Обща схема на действие: Мрежовият слой на абоната А подава данни на каналния слой на абоната А, които трябва да достигнат до мрежовия слой на абоната В, ако А и В са директно свързани, или до мрежовия слой на маршрутизатор С, ако не са. Каналният слой на А кадрира според съответния протокол данните и ги предава на физическия слой на А; физическият слой на А ги предава на физическия слой на В (или на С); физическият слой на

В ги предава на каналния слой на B; извършва се проверка на контролната сума на кадъра и, ако няма грешка, кадърът се предава на мрежовия слой на B, а, ако има грешка, кадърът се отхвърля. Аналогично се извършва предаването на кадри между маршрутизатори.

Обикновено данните на канално ниво се обменят на порции с фиксирана дължина, наречени **кадри**. В мрежите с пакетна комутация (такива, при които съобщенията се разделят на пакети; всеки пакет се предава индивидуално в комуникационната подмрежа и възловият компютър се грижи за съхраняването и предаването не на единици-съобщения, а единици-пакети; други видове комутация – на канали, където се установява физически канал за обмен на информация и по този канал се предава едно съобщение; на съобщения, където всяко съобщение се изпраща в комуникационната подмрежа, която избира неговия маршрут до назначението му) обикновено кадрите съвпадат по дължина с размера на пакета. Форматът на кадрите се определя от избрания протокол за предаване на канално ниво.

Кадърът е пакет, който каналният слой използва при обмена на информация с физическия. Физическият слой възприема информацията от каналния слой като поток от битове. Кадърът е поредната част от този поток, която каналният слой контролира за наличието на грешки. Формирането на кадрите се осъществява, като информационният поток, получен от мрежовия слой, се дели на части, към всяка от които се добавя служебна информация. Обратно – в посока от битове от физическия слой, каналният слой идентифицира (разпознава начало и край) кадрите, проверява ги за грешки (при наличие на такива може да ги коригира, зависи от конкретната реализация), премахва се служебната информация на каналния слой и информационният поток се предава към мрежовия слой. Възниква въпросът как да стане разделянето на кадри. Понастоящем се използват три метода:

- Броят се отделните символи, като в заглавието на кадъра се указва и броят им (и символите от заглавната част участват в броенето). Проблемът е, че може поради грешка да не може да се установи точно размерът на даден кадър, а от там и на всички останали.

- Началото и краят на всеки кадър се определят от уникална символна последователност (STX и ETX); start – DLE STX; end – DLE ETX. Ако в информацията на кадъра се срещне DLE, то се удвоява. Проблемът е, че методът е базиран на ASCII кодирани 8-битови символи, което е свързано с нерационално използване на възможностите на комуникационната среда.

- Всеки кадър започва и завършва със специална последователност 01111110, наречена флагов байт. В съдържанието на кадъра след всяка последователност от 11111 се добавя 0.

В много протоколи се среща комбинация от първите два или от първия и третия начини.

Предаване – За да се избегнат грешките на канално ниво, се използват два механизма: потвърдено обслужване (потвърждаване на всеки получен кадър) и установено обслужване (номериране на последователностите от изпращани кадри). Три са основните варианти на комуникационната услуга, която каналният слой може да осигурява:

- Непотвърдено неустановено обслужване – отсъства обратна информация от B или C за приемането на кадрите; няма установяване на връзка и освобождаването й след това => не се държи сметка за последователността, в която се получават кадрите; използва се, когато навременното регулярно получаване на кадрите е по-важно от тяхната достоверност, например при предаване на реч или видео.

- Потвърдено неустановено обслужване – всеки кадър се потвърждава; по принцип потвърждаването се прави на ниво транспортен слой, но то се отнася до последователност на пакети; потвърждаване на канално ниво се налага при ненадеждна комуникационна среда, например безжична.

- Потвърдено установено обслужване – характеризира се с три фази: установяване на връзката със заделяне и инициализация на необходимите ресурси; предаване на кадрите; освобождаване на връзката и заделените за нея ресурси => така се гарантира и предаването на кадрите, и последователността, в която те се предават.

Грешки:

- Кадърът пристига в B с грешка -> B изпраща на A потвърждаващ кадър, в който съобщава за грешката; A изпраща кадъра отново. Ако не е налице потвърдено обслужване, A изпраща кадъра след определен TIMEOUT > от времето за потвърждаването.

- Кадърът не пристига в B -> след изтичане на TIMEOUT A изпраща кадъра отново.

- Кадърът е пристигнал в B, но потвърждението се е загубило -> A изпраща кадъра отново след TIMEOUT, B получава и новия кадър и се получава дублиране -> присвояват се номера на всеки кадър от изпращаните (установено обслужване), така че B да различава оригинала от дубликатите.

- A изпраща кадри по-бързо, отколкото B може да ги приеме -> въвеждат се механизми за управление на потока от кадри, които осигуряват обратна информация на A за темпа на предаване.

За установяване на състоянието на канала в A се прави настройка по брой повторения n. След като A предаде един и същи кадър n пъти, A счита канала за неработещ и прекратява опитите за изпращане на кадъра.

При препредаване на кадри не е нужна корекция на грешка, а само установяването ѝ.

Какво е характерно за Ethernet.

Локалната мрежа Етернет използва метода за достъп МДОН/РК (Множествен достъп с откриване на носеща [честота] и разпознаване на конфликтите). Същността на метода се състои в непрекъснато подслушване на канала – каналът се подслушва преди предаване на данни и данни не се предават, докато той не се освободи. След като каналът се освободи, данни не се предават още определено време >= прозореца за конфликти. Така цялата съобщителна среда преминава в пасивно състояние. Предаването започва, но подслушването продължава. Ако се открие интерференция в приемания сигнал, това означава, че е настъпил конфликт. След установяването на конфликта предаването продължава още известно време = прозореца за конфликти. Станцията, участвала в конфликта, отлага препредаването на пакета със случайно подбран интервал от време.

Основно предимство на МДОН/РК е, че предаващата станция със сигурност знае кога пакетът е предаден успешно в канала. Другото предимство е малкото време, в което каналът стои зает по време на конфликт. Основният недостатък е силната зависимост от времето за разпространение на сигнала.

Протоколът на поднивото за достъп до средата (MAC – Media Access Control) в Етернет стандартизира следния формат на кадъра:

7 B	1 B	6 B	6 B	2 B	0-1500 B	0-46 B	4 B
Преамбюл	Начало на кадъра / Начален разделител	Адрес на получателя	Адрес на източника	Дължина на данните	Данни	PAD	Контролна сума

- Кадърът започва със синхронизиращи байтове преамбюл – поредица от 10101010.
- Началният разделител / Началото на кадъра (SOF, Start of Frame) е поредицата 10101011.
- Най-старшият бит на адреса на получателя е 0 за нормален адрес и 1 за групов адрес (multicast). Адрес, съставен само от 1, е broadcast адрес. Бит 46 се използва за различаване на локални от глобални адреси.
- Полето за дължина на данните сочи броя байтове в следващото поле за данни + PAD. Кадри, по-къси от 64 байта, създават проблеми, затова минималната дължина на кадъра (от адреса на получателя до контролната сума включително) е 64 байта => трябва да има поне 46 байта данни, а, ако няма, се запълва с PAD до 46 байта.
- Контролната сума се използва, за да се следи за това дали кадърът е коректно получен.

Предаването на къс кадър е проблемно, тъй като е възможно, преди достигането на първия бит до края на кабела, друга станция да започне предаване. Тогава възниква конфликт. По-близката до конфликта станция разбира за конфликта, прекратява предаването и генерира 48-битова поредица (jam), с което предупреждава всички станции за конфликта. Ако обаче другата станция, която е участвала в конфликта, вече е прекратила предаването, тя е счела, че кадърът е успешно предаден, и го е изчистила от буферите си преди jam сигналът да достигне до нея. Така кадърът е загубен. За предотвратяването на това времето за предаване на кадър трябва да е повече от 2 пъти времето за разпространение на кадър от единия до другия край на кабела.

Статична маршрутизация – таблица и избори.

Основна функция на мрежовия слой е доставянето на пакети от източника до местоназначението им. Изпълнението на тази задача може да изисква няколко на брой стъпки, в които пакетите преминават последователно през различни междинни маршрутизатори. Всеки от маршрутизаторите избира подходяща следваща стъпка за предаване на пакетите въз основа на записаната в неговата **маршрутна таблица** информация. В маршрутната таблица обикновено са записани само част от всички възможни следващи стъпки към дадено местоназначение. Съществуват различни критерии за избор на най-добър път, например – брой на стъпките, закъснение на пакетите, пропускателна способност, натоварване, надеждност или цена на връзките. С всеки маршрут е асоциирана **метрика**, която представлява функция на една или повече от тези променливи.

Маршрутизацията се нарича **статична**, ако маршрутната таблица е попълнена ръчно от администраторите. Тя е удобна единствено в малки мрежи, при които рядко настъпват промени. Полезното ѝ е, че е предсказуема, но пък изисква много обстойно планиране и не се адаптира динамично към промени в топологията на мрежата.

Централизирана маршрутизация – недостатъци.

При **централизираната маршрутизация** маршрутизирането се извършва чрез специален управляващ мрежата компютър (маршрутен управляващ център (routing command center)),

който изчислява конфигурацията на мрежата и построява маршрутната таблица за всички възли в мрежата. Периодично се препостроява тази маршрутна таблица и се изпраща към възлите в мрежата. Основно предимство е простотата на този модел – решение за маршрутизацията се взима самостоятелно от един компютър. Освен това, информация като параметри на мрежата и на връзките в нея, които не се променят често, могат да се пазят в централна база от данни и няма нужда да бъдат обявявани всеки път.

Недостатъци на централизираната маршрутизация:

- Ако се случи нещо с управляващия компютър, инструкциите за маршрутизиране не могат да бъдат променяни, докато не се поправи този компютър.
- Маршрутната таблица не отчита претоварванията в мрежата.
- Когато се промени маршрутната таблица, се хабят мрежови ресурси за разпращането на обновената таблица на всички възли в мрежата.
- Ако мрежата се разцепи на две, едната част остава без централен маршрутизиращ компютър и тази част от мрежата не може да се адаптира към промени в топологията.

Разпределена маршрутизация.

Разпределената маршрутизация позволява всички възли в мрежата да взимат свои решения за маршрутизирането, следвайки формален протокол за маршрутизиране. При разпределената динамична маршрутизация маршрутизаторите самостоятелно и автоматично определят следващите стъпки към всички известни направления и реагират своевременно на евентуални промени в топологията на мрежата, отпадането на устройства или връзки между тях. При възникнали промени, времето, необходимо за преизчисляване на маршрутните таблици и достигане до непротиворечиво описание на новата топология, определя скоростта на сходимост на маршрутизиращия протокол.

Има два основни типа маршрутизиращи протоколи:

- **Протоколи с дистантен вектор/вектор на разстоянието** – маршрутизаторите обменят информация за топологията на цялата мрежа само със своите съседи. Тези протоколи изискват по-малко мрежови и изчислителни ресурси, но имат по-ниска скорост на сходимост и не гарантират отсъствието на цикли в маршрутите. Сериозен недостатък е, че добрите новини (включването на нов маршрутизатор) се разпространяват бързо в мрежата, но лошите новини (изключване на маршрутизатор) обикновено изискват твърде голям брой периодични съобщения, за да достигнат до всички маршрутизатори. Всеки ред в маршрутната таблица съдържа адрес на дадено местоназначение, адрес на следващата стъпка към това местоназначение и метрика. Предполага се, че всеки маршрутизатор знае метриката на връзките до своите съседи. Метриката може да бъде „брой стъпки", „пропускателна способност", „цена на връзката", „натоварване", „надеждност" или „закъснение на пакетите". Действието е следното: през определен интервал от време всеки маршрутизатор изпраща на своите съседи съдържанието на маршрутната си таблица; когато маршрутизатор получи таблицата на свой съсед, той преизчислява своята.
- **Протоколи със следене състоянието на връзката** – маршрутизаторите обменят информация само за своите връзки към мрежата с всички маршрутизатори в нея. Тези протоколи имат по-висока скорост на сходимост и предотвратяват възникването на цикли в маршрутите, но с цената на повече процесорно време и памет, както и увеличен обем на

обменяната между маршрутизаторите информация и увеличена сложност на протоколните реализации. Действието е следното: маршрутизаторът изследва първо своите съседи, като им изпраща ехо-пакети, а те му отговарят с идентификационните си номера (имена) и със служебните си адреси; прави изчисления, изследва околната мрежа; създава специални пакети, които съдържат вече получената информация за връзките със съседите; изпраща тези пакети до всички маршрутизатори в мрежата; след като получи всички пакети, строи граф на мрежата и по алгоритъма на Дейкстра изчислява най-кратките пътища до всички маршрутизатори в мрежата. Тъй като всеки трябва да получи всичко, се използва алгоритъм с наводняване.

IP дейтаграма.

Интернет протоколът (IP) дефинира основната единица за предаване на данни в интернет – **дейтаграмата**. Най-общо, терминът „пакет" се използва за всякакво съобщение, оформено като пакет, докато „дейтаграма" се използва в случаите на ненадеждна услуга.

IP (Internet Protocol v4) е протокол на мрежовото (3-то) ниво, който използва дейтаграмен (ненадежден) метод на предаване без установяване на връзка. По тази причина всяка дейтаграма трябва да съдържа пълна информация за адресите на получателя и източника. IP не гарантира успешното получаване на дейтаграмите в местоназначението. Ако за дадени приложни програми е необходима надеждност при предаването, то тя се гарантира от протоколите от по-високо ниво (например TCP). Основните функции на IP са: адресиране (чрез заглавната част на дейтаграмата се задават адреси, чрез които междинните маршрутизатори избират път за пакета); фрагментиране (позволява се големи по размер пакети да преминават през мрежи, които могат да обработват само малки пакети); таймаут на пакетите (стойността на полето „време на живот" – TTL, се инициализира от изпращача и се намалява с 1 всеки път, когато пакетът премине през маршрутизатор; така се предотвратява евентуално зацикляне); тип на услугата (може да се задават приоритети на трафика) и опции (изисквания за пътя и проследяване на пътя).

ФОРМАТ НА ДЕЙТАГРАМАТА ЗА ПРОТОКОЛ IP ВЕРСИЯ 4:

<table>
<tr><td colspan="4">0 15 16 31</td></tr>
<tr><td>Версия – 4 бита</td><td>Дължина на заглавната част – 4 бита</td><td>Тип на услугата – 8 бита</td><td>Обща дължина на дейтаграмата в байтове – 16 бита</td></tr>
<tr><td colspan="3">Идентификатор – 16 бита</td><td>Флагове – 3 бита</td><td>Отместване на фрагмента – 13 бита</td></tr>
<tr><td colspan="2">Time-To-Live – 8 бита</td><td>Протокол – 8 бита</td><td colspan="2">Контролна сума на заглавната част – 16 бита</td></tr>
<tr><td colspan="5">IP адрес на източника – 32 бита</td></tr>
<tr><td colspan="5">IP адрес на получателя – 32 бита</td></tr>
<tr><td colspan="5">Опции (не е задължително да присъстват, при необходимост се добавят 0)</td></tr>
<tr><td colspan="5">Данни (не е задължително да присъстват)</td></tr>
</table>

- В зависимост от версията дейтаграмата има различен формат. В случая на IPv4 стойността на полето за версия е 4.

- Дължината на заглавната част (header length) е броят 32-битови думи, вкл. полето за опции. Минималната стойност е 5, а максималната – 15, т.е. може да има до 40 байта опции.

•Полето за тип на услугата (type of service) се състои от 3 водещи бита, 4 бита за вид на услугата и 1 неизползван бит, който трябва да е 0. 4-те бита за вид на услугата задават съответно изисквания за: минимум закъснение, максимална пропускателна способност, максимум отказоустойчивост, минимум цена, като наведнъж само един от тези 4 бита може да бъде 1.

•Максималната стойност на общата дължина на дейтаграмата е 65,535 байта. Това поле може да бъде променено при фрагментиране на дейтаграмата.

•Полето идентификатор има еднаква уникална стойност за всички фрагменти, принадлежащи на дадена дейтаграма.

•Старшият бит на полето флагове (flags) е резервиран. Ако следващият е 1, не трябва да се извършва фрагментация на дейтаграмата. Най-младшият бит е 1 за всички фрагменти от една дейтаграма с изключение на последния.

•Отместването на фрагмента (fragment offset) указва местоположението на фрагмента в дадена дейтаграма. Дължината на всички фрагменти без последния трябва да е кратна на 8 байта (елементарен фрагмент).

•Времето за живот (Time-To-Live) е горната граница на броя маршрутизатори, през които може да премине дейтаграмата.

•Протокол (protocol) – указва с кой протокол от по-високо ниво се работи.

•Контролната сума (header checksum) се пресмята само върху полетата на заглавната част на IP дейтаграмата: 1. Първоначалната стойност се инициализира да бъде 0; 2. Заглавната част се разглежда като последователност от 16-битови думи в допълнителен код. Изчислява се сумата (побитова сума по модул две) на тази първоначална стойност и последователността от думи на заглавната част; 3. Допълнението до 16 бита единици на така изчислената сума се записва в полето за контролна сума.

IP адресация – класова, безкласова.

Адресите в IPv4 са 32-битови двоични числа, които е прието да се записват като 4 десетични числа, разделени от точки (всяка от тези 4 части се нарича октет). Логически мрежовите адреси имат 2 части: мрежов идентификатор (netid) и идентификатор на хоста (hostid). Ако hostid е само от 0, това е адресът на мрежата, към която хостът принадлежи. Ако пък е изцяло от 1, това е адрес за предаване до всички машини (broadcast address), който едновременно адресира всички машини в дадена мрежа. Например 149.76.255.255 адресира всички хостове в мрежа 149.76.0.0. Има и 2 резервирани адреса – 0.0.0.0 (маршрут по подразбиране, използва се при маршрутизацията на IP дейтаграмите) и 127.0.0.0 (резервиран за локален IP трафик; присвоява се интерфейс за обратна връзка на хоста – loopback interface). Адресът 127.0.0.1 (може и 127.x.x.x, но за удобство пишем 127.0.0.1) е резервиран и се използва за служебен адрес на самия хост (localhost address).

Адресни класове в Интернет:

A	0	network (1+7)			host (24)		
B	1	0	network (2+14)			host (16)	
C	1	1	0	network (3+21)			host (8)
D	1	1	1	0	multicast address (28)		
E	1	1	1	1	reserved for future use (27)		

нямат адреси на хостове

Съществуват 128 мрежи от клас А и над 16 милиона хоста за всяка мрежа от този клас. Т.е. има малко мрежи с много хостове. Адресите от клас А са предназначени за много големи мрежи, тези от клас В са за средни по размер мрежи (16 384 мрежи от клас В и над 64 000 хоста за всяка от тях), а от клас С са за малки мрежи.

При клас С: мрежите 192.0.0.0 и 223.255.255.0 са резервирани за служебно ползване.

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

При адресите от клас D последните 28 бита се използват за адрес за едновременно предаване до група машини (multicast address). Използването им позволява дадена IP дейтаграма да се предаде до „група от хостове". Например едновременно предаване до група хостове се използва за мрежови видео и аудио конференции и за LAN TV.

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

Преобразуване на IP в MAC и обратно.

За адресация в IPv4 се използват 32-битови IP адреси, а хостовете, свързани към Етернет, притежават 48-битови хардуерни/MAC адреси (които се записват в 16-ичен вид). За **установяване на съответствие между IP адреса и MAC адреса** се използва **ARP** (Address Resolution Protocol). Когато даден хост иска да изпрати дейтаграма към машина в локалната мрежа, знаейки IP адреса, но не и Етернет адреса, той изпраща ARP пакет-заявка, от тип broadcast. Търсеният хост изпраща ARP съобщение до подателя с информация за Етернет адреса си, а всички останали хостове игнорират заявката.

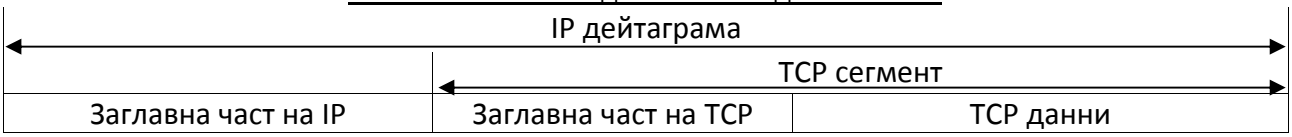
Протоколът **RARP** (Reverse Address Resolution Protocol) се използва в обратната посока, за **намирането на съответствие между Етернет адреси и IP адреси**. Действието на RARP се основава на наличието на уникален хардуерен Етернет адрес за всяка система в локалната мрежа. При инициализиране на машината без твърди дискове RARP протоколът прочита този адрес от интерфейсната карта и предава до всички станции в мрежата пакет-заявка. RARP сървърът отговаря на тази заявка, като в пакета-отговор се съдържа IP адресът, съответстващ на изпратения Етернет адрес.

TCP формат.

TCP (Transmission Control Protocol) е протокол на транспортното (4-то) ниво, създаден специално да предоставя надеждни транспортни услуги за потока от байтове на

протоколите от по-горните нива въпреки ненадеждността на използваната мрежова среда. Функциите на този протокол се осъществяват от програмен модул, който в общия случай е част от ядрото на операционната система. TCP установява логическа връзка „от край до край" между 2 приложни програми. TCP осигурява възможност за двупосочно предаване на данните (пълен дуплекс), както и за надежден обмен на поток от номерирани байтове. TCP се справя с проблеми като загубване на пакети, дублиране на пакети и получаване на пакети не в реда в който са изпратени.

ОПАКОВАНЕ НА TCP ДАННИТЕ В IP ДЕЙТАГРАМА:



Обменът на информация се извършва посредством сегменти. При предаване TCP получава данни от по-горния слой, разделя ги на части, опакова ги в „сегменти" и ги изпраща на IP протокола. Той от своя страна опакова сегментите в дейтаграми и извършва маршрутизирането на всяка им. При приемане IP разопакова пристигналите дейтаграми, след което предава получените сегменти на TCP, който сглобява и подрежда данните от сегментите в съобщения към по-горните слоеве така, както те са били изпратени.

Всеки край на TCP връзката се идентифицира с IP адреса на хоста и със 16-битов номер на порт, който определя програмата, използваща тази връзка. Двойката „адрес на хост + номер на порт" се нарича гнездо (socket). Комбинацията от гнездата на източника и на получателя е уникална и идентифицира TCP връзката. Това позволява едно гнездо да се използва едновременно от няколко TCP връзки, т.е. да се мултиплексира.

Заглавната част на TCP сегмента включва задължителни полета с фиксиран размер от 20 байта (5 32-битови думи), към които може да бъде добавено поле *опции*, след които може да има поле на обменяните данни.

ЗАГЛАВНА ЧАСТ НА TCP СЕГМЕНТ:

<table>
<tr><td colspan="8">0 15 16 31</td></tr>
<tr><td colspan="7">Номер на порта на източника – 16 бита</td><td>Номер на порта на получателя – 16 бита</td></tr>
<tr><td colspan="8">Пореден номер (Sequence number) – 32 бита</td></tr>
<tr><td colspan="8">Номер на потвърждението (Acknowledgement number) – 32 бита</td></tr>
<tr><td>Дължина на заглавната част – 4 бита</td><td>Резерви-рани – 6 бита</td><td>U R G</td><td>A C K</td><td>P S H</td><td>R S T</td><td>S Y N</td><td>F I N</td><td>Размер на прозореца – 16 бита</td></tr>
<tr><td colspan="7">Контролна сума на TCP (checksum) – 16 бита</td><td>Указател за спешни данни – 16 бита</td></tr>
<tr><td colspan="8">Опции (не е задължително да присъстват)</td></tr>
<tr><td colspan="8">Данни (не е задължително да присъстват)</td></tr>
</table>

Номерата на портовете на източника (source port*)* и на получателя *(*destination port*)*, заедно със съответните IP адреси, образуват двете гнезда, идентифициращи връзката. Следва поредният номер на първия байт, записан в полето *данни* на този сегмент. Номерът на потвърждението е номерът на първия байт данни, който се очаква да се получи със следващия сегмент. Дължината на заглавната част *(*header length*)* се мери в брой 32-битови думи. Фактически с него се определя началото на полето данни и затова се нарича и Data Offset. Ако някой от следните 6 еднобитови флага е установен, т.е. има стойност 1, то:

- *URG* – валиден е указателят за спешни данни (urgent pointer), т.е. трябва да се преустанови обработката на получените данни, докато не се обработят байтовете, към които сочи указателят за спешни данни;
- *ACK* – валиден е номерът на потвърждение;
- *PSH* (push) – наличните данни трябва да се изпратят възможно най-бързо към техния получател, т.е. източникът не изчаква образуването на пълен сегмент и съответно получателят не чака запълването на приемния буфер;
- *RST* (reset) – ако е 1, то сегментът служи за прекратяване на TCP връзката;
- *SYN* (synchronization) – сегментът се използва при установяване на TCP връзката и за изпращане на началния номер (задава се в полето *пореден номер)*, от който ще бъдат номерирани байтовете, т.е. се иска синхронизиране на номерацията на сегментите;
- *FIN* (finish) – изпращачът прекратява предаването на данни.

Размерът на прозореца (window size) определя колко байта могат да бъдат изпратени и съответно приети наведнъж, без препълване на входния буфер. Указателят за спешни данни (urgent pointer) указва позицията на първия байт на спешните данни спрямо началото на полето данни. Контролната сума (checksum) се изчислява върху целия TCP сегмент и осигурява проверка за коректността на данните. При нейното пресмятане участва и т.нар. псевдо-заглавна част (pseudoheader) – 12 байта, включващи някои полета от заглавната част на IP дейтаграмата.

ПСЕВДО-ЗАГЛАВНА ЧАСТ НА TCP ДЕЙТАГРАМА:

0	15	16	31
IP адрес на източника – 32 бита			
IP адрес на получателя – 32 бита			
Нула	Протокол (06 за TCP)	Дължина на TCP сегмента	

Ако няма *опции* или са по-малко от 32 бита, останалите битове се запълват с 0 (padding).

TCP – 3-way hand shake.

Първоначално отваряне на връзката (Connection Establishment Protocol): Необходимо е всеки един от двата хоста да изпрати на другия началния номер (initial sequence number) на байтовата последователност, която ще изпраща, и съответно да получи насрещното потвърждение за получаването на този номер. Процедурата е следната:

1. Клиентът изпраща SYN сегмент, в който задава и номера на порта на сървъра, както и началния номер на потока байтове (x).
2. Сървърът отговаря със собствен SYN сегмент, включващ началния номер на неговия поток от байтове (y). Чрез този сегмент сървърът изпраща и ACK за потвърждение на SYN сегмента на клиента.
3. Клиентът потвърждава получаването на този SYN сегмент от сървъра със сегмент ACK.

При обмена и потвърждението на началните номера на байтовите последователности от всеки хост са необходими общо три стъпки, при които се обменят съответно 3 сегмента. Затова тази процедура се нарича **диалог с три съобщения** (**three-way handshake**):

Стъпка	Клиент	Сървър
1	SYN = 1, SEQ = x	
2		SYN = 1, SEQ = y, ACK = x + 1
3	SEQ = x + 1, ACK = y + 1	

TCP – разлики с UDP.

UDP (User Datagram Protocol) е прост транспортен протокол за предаване на дейтаграми в мрежите с комутация на пакети. За разлика от TCP, UDP не осигурява надежден транспорт и подредба на пакетите, не се справя със загубени или дублирани пакети. Дейтаграмите се изпращат от източника, без да се контролира дали са пристигнали до получателя. Услугите, които UDP добавя върху услугите на IP са мултиплексиране на логическия канал за връзка между двата хоста, както и откриване на грешки и контролна сума на данните. UDP, за разлика от TCP, може да се използва за multicast.

HTTP.

HTTP (Hypertext Transfer Protocol) е протокол на приложното (7-мо) ниво. Той представлява прост текстов протокол, който се използва от услугата WWW за осигуряване на достъп до практически всякакъв вид данни, наричани събирателно ресурси.

При HTTP протокола има понятия като клиент (обикновено това са Web-браузърите) и сървър (това са Web-сървърите). HTTP най-често се използва с TCP/IP, но практически може да работи върху всякакъв протокол, който предоставя надежден транспорт. Обикновено HTTP протоколът работи върху стандартен TCP сокет, отворен от клиента към сървъра. Стандартният порт за HTTP е 80, но може да се използва и всеки друг TCP порт.

Комуникацията по HTTP се състои от заявка (request) – съобщение от клиента към сървъра, и отговор (response) – отговор на сървъра на съобщението от клиента. HTTP заявките имат 3 основни елемента: метод на достъп, Request-URI и header-полета.

Методът описва вида на HTTP заявката, изпратена от клиента. Най-често използваните методи са GET и POST. Чрез GET клиентът изисква някакъв ресурс от Web сървъра, а POST служи за предаване на данни към сървъра и извличане на ресурс. Идентификаторът

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