Мрежи

**OSI модел – най-обща характеристика на нивата.**

Стандартът OSI (Open System Interconnection), разработен от Международнатаорганизация 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.

**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 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.х.х.х, 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)

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.х.х.х от клас А, 16 мрежи 172.16.x.x до 172.31.x.x от клас В и 256 мрежи 192.168.х.х 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-aта. 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.

**TCP формат.**

**TCP (Transmission Control Protocol)** е протокол на транспортното (4-то) ниво, създаден специално да предоставя надеждни транспортни услуги за потока от байтове наcпротоколите от по-горните нива въпреки ненадеждността на използваната мрежова среда.

Функциите на този протокол се осъществяват от програмен модул, който в общия случай еcчаст от ядрото на операционната система. TCP установява логическа връзка „от край до край” между 2 приложни програми. ТСР 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.

**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 (х).

2. Сървърът отговаря със собствен SYN сегмент, включващ началния номер на неговия поток от байтове (у). 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):