1. Компютърни мрежи. Апаратура и топологии.

През 60-те се създава проектът ARPA , за създване на мрежа м/у уневирситетите. С течение на времето се появавя нуждата да се свързват галеми центрове в мрежа и така с развиването на arpa проектът се стига до реализирането на internet.

В началото компютърните технологии са били свързани с телефонията, т.к. са използвале вече готовата и мрежа. При телефоните говорът се предава като трептения с определена честота. Микрофонът преобразува въздушното трептене в електрическо трептене със същата честота, след което сигналът преминава през кабелита като слаб ток. На другия край се възстановява трептенето.

Регионалните централи са свързани помежду си с магистрали, които са телефонно организирани. Свързването се базира на комутация на каналите. Не се очаква всички абонати да бъдат винаги активни. Използват се временни логически канали по време на разговора между два абоната.

За да се уплътни тел трафик се предават много разговори по един проводник като се използва модулация(амплитудна, фазова,честотна).

Тел с-ми са създадени за предаване на аналогови сигнали, затова се използват специални цифрови модеми за предаването на ел сигнали.

ADSL модема не пречи на използването на телефона. Той слага допълнителен сигнал след тел и го маха преди достигането на АТЦ-то.

За да се увеличе пропусклевоста на канала кодавите таблице се опростяват – букви, цифри, преп знаци в 7 бита. Буквено цифровите терминали са работели строго само с тези днаци и са биле поредови (cr – е бил важен знак).

На базата на тези тел с-ми се развива отдаленият достъп до компютрите.

Комп. Мрежа есъвкупност от компютри свързани помежду си с комуникационна подмрежа. Тя представлява мрежа от комун канали и апарати които ги обслужват. Host- компютрите в мрежата които приемат и извършват същинската работа. Терминали – нямат собствени изчислителни процеси, но използват тези на хостовете.

Когато използваме браузъра имеме роля на терминал, а когато пуснем сървър приемащ заявки- хост.

При мрежите със селекция:

- много бърз общ канал, през който преминават всички данни, всеки компютър отделя данните предназначени за него от общия поток

При мрежите с маршрутизация:

- хостовете са свързани към специални възли, които осъществяват разпознаването на входящия поток, доставят на съответния получател само определените за него данни
- много потоци от данни, които се разклоняват и разпределят от всеки възел

От гледна точка на физическите характеристики(разстоянието м/у хостовете) мрежите се класифицират на 1м – комп. Система, 10м – стая – LAN, 100м – сграда – LAN, 1 км – кампус – LAN, 10км – градска – MAN, >1 нас място – WAN, Цялата земя - INTERNET

Технически е възможно да се направи високоскоростен канал, затова за LAN се използа мрежа със селекция.т.к. броя на абонатите е ограничен и няма да се задръсти канала. В WAN няма ограничение на абонатите и затова се изграждат с машрутицация. Интернет също е с маршрутизация. MAN е хибрид м/у lan и wan мрежите. В тях се вкарват технологии и от двата вида.

2. Структура и еталонен модел на мрежите. Характеристика на нивата.

Проектирането и изграждането на мрежите трябва да става на нива т.к. създаването на едно цяло се оказва невъзможно. Всеки слой си има своя функционалност в рамките на мрежата. За взаимодействието м/у два съседни слоя се използва интерфейс, като долния слой обслужва заявката на горния. Във свеси слой има набор от програми които се обръщат към програми от долния слой. Основната задача на мрежите е иминно това взаимодействие. То става чрез протоколи които са нещо като правила как да комуникират съответните слоеве. Всяко ниво си следи собствената структура от данни, без да се интересува дали пренас, служебна част ими данни на горното ниво.Мрежите се състоят от 7 функционални слоя. Като всеки абонат има тези нива.

Физическото ниво осъществява обмяната на сигнали. Интерфейсът му е за предаване на безкрайна серия от битове. Тук се решава как точно да се предават битовете чрез сигнали, най-често това става с ел ток. Физическия сигнал може да е импулсен и чрез импулсни сигнали да се предават 0 и 1. Пример за такъв интерфейс е модемният RS232. Пренега битовете се предават на крайното устройство което генерира сигналите. Носещият сигнал е синусоида с носеща честота, в/у която се наслагват 0 и 1 чрез модулация.

Каналното ниво има задачата да структуира битовете, които се предават към друг абонат, така че той да може да различе началото и края. Въвеждат се кадри, които са порция от битове която се приема за вярна в отсрещното канално ниво. Данните които се предават остават непроменени, кадрите се различават в служебната си част.

Мрежовото ниво се грижи за формиравнето на ком. мрежа, в която даден пакет може да премине м/у няколко приемопредавателя, за да стигне до краен получател. На мрежовото ниво ме се дават данни и то трябва да направи пакет и да то прати на получателя. Данните пристигат от горното ниво а в това им се слага хедър. Мрежооото ниво е необходимо при необходимост от маршрутизеране.

Транспортното ниво получава произволна дълго съобщение, нарязва го на по малки пакети и го предава на мрежовото ниво. Чрез транспортния протокол се проверява дали съобщението е вярно.

Сесийното ниво се грижи за това два абоната да извършват взаимодействие в рамките на една сесия. Сесията може да е м/у два или повебе абоната. Задачата на сесията се установява при започването й, тя изяснява структурата на взаимодействието(дължина на съобщението и др). В рамките на сесията абонатите си изпращат съобщения.

Представителното ниво е измислено за да реши проблема с отворените системи – абонатите да работят с различни компютри. Тук символите се преобразуват в Unicode и така в мрежата пътуват едни и същи данни. Също така това ниво се грижи за шефроването(при прихващане на съобщение то да се разбере) и за автентикацията(при получаване да се удостовери автентикацията) на данните. Представителното ниво има за задача да разпознава рачличните типове данни.(уеднакрявяне на файловете практически нямаме).

Приложното ниво осъществява връзката с приложението. То дава интерфейс на приложенията към мрежите. И т.к. приложенията са различни иаме специфични интерфейси, които комуникират един с друг.

3. Видове мрежи – комутация на канали, пакети и съобщения

Мрежите със селекция:

- много бърз общ канал, през който преминават всички данни, всеки компютър отделя данните предназначени за него от общия поток

Мрежите с маршрутизация:

- хостовете са свързани към специални възли, които осъществяват разпознаването на входящия поток, доставят на съответния получател само определените за него данни
- много потоци от данни, които се разклоняват и разпределят от всеки възел

В комп подмрежа се слагат точки за получаване и препращане на съобщения – възли. Абонатите в мрежата в рамките на сесията си обменят съобщения, т.к. няма точен регламонт за какво се ползва мрежата съобщенията носят цялата информация и не могат да бъдат ограничавани. Възелът запомня съобщението изцяло в буфера си и тогава го предава нататък. Така то преминава през два възела. Като ще бъде изтрито от първия едва след получаването на потвърждение за получаване от другия възел. Дотогава възелът пази своето копие.

Комутация с пакети. Съобщенията могат да бъдат много големи, затова те де пазят при автора а той от своя страна може да нареже съобщението на парчета с предварително зададена големина. Така възлите могат да знаят колко точно пакета могат да пазят в паметта си. Също така по този начин се осигурява тряфик на пякети с еднаква големина из ком подмрежа. Създават се устройства за бързо предаване и преработка на съобщенията. Оригиналът се пази в източникът. А приемникът ако успее са сглоби съобщението праща потвърждение. Съща така може да поиска и само отделни части от съобщението които са се загубили или не са валидни. Преминаването на пакетите през ком подмрежа е чриз датаграми и общ канал. При датаграмите пакетите имат номер и съответната информация. Тькь обаче получателят ги получава в разбъркан ред. Плюс е това че всеки датаграм може да се изпраща в различна посока, а минуса че могат да се загубят по пътя. При друхия подход преди изпращането на първия пакет се установява един общ временнен канал за всички пакети. Физически пакетите се движат отделно един от друг но се движат по един и същи канал и пристигат в хронологичен ред. Така е по лесно да се възстановят загубите.

И при двата вида комутация ком подмрежа ще е съвкупност от рутери и ком канали които ге свързват. Обменът на данни е на няколко нива – на ниско ниво(м/у 2 рутера) до високо ниво(м/у 2 абоната). Каналите могат да са симплексни (само в едната посока), полудуплексни(в едната посока но посоките се редуват) и дуплексни(едновременно в двете посоки).

При мрежите със селекция логиката е свързана с физическата реаличация. При общата щина от абоната тръгват съобщения към двата края и абонатите за които пакетита са предназначени ги коперат в себе си. Каналът е бърз и големината на съобщенията може да е различен, но ако са големи те се нарязват на по малки. Тук няма мяршрутезяция и неможа де се разбера дали съобщението е получено. При пръстен- данните се двежат в едната посока и преминават през всички абонати. Един варянт е анд събщението е прочетено, това да се отбележи в хедъра, и връщайки се в изпращача той да разбира дали всичко е ок. А другия варянт е абонатът който получи съобщението да не го препредава. Така ако изпращача си полечи съобщението, то тогава то не е достигнало до където трябва.

4. физическо ниво в мрежите. Теоритически основи и среди за предаване.

Физическото ниво осъществява обмяната на сигнали. Интерфейсът му е за предаване на безкрайна серия от битове. Тук се решава как точно да се предават битовете чрез сигнали, най-често това става с ел ток. Физическия сигнал може да е импулсен и чрез импулсни сигнали да се предават 0 и 1. Пример за такъв интерфейс е модемният RS232. При него битовете се предават на крайното устройство което генерира сигналите. Носещият сигнал е синусоида с носеща честота, в/у която се наслагват 0 и 1 чрез модулация. Тя е удобна защото честотата и фазовия ъгъл не се променят, само амплетудата намалява.

В канал без шумове макс скорост на данните е max datarate=2H.logV, а със шумове =Hlog(1+ S/N) , H-честотната ширина на канала, V-скороста, S- сигнал, N- шума,

Среди за пренасяне. Първата среда за пренасяне на данни са били перфокартите, слид това магнитните ленти. По късно за пренасяне на ел сигнал започват да се използват проводнице – усукани двойки. Колкото по-гъсто се усучат жиците толкова устойчивостта им към щумове е по голяма. Медните жици които са изолирани и след това усукани са 3 и 5-та категория(по-гъста). 4 усукани успоредни двойки с обща обвифка образуват UTP(unshielded twisted pair). Ако обвием тези двойки с тънка мрежа от тънък меден проводник, отгоре сложим пластм. обвифка и зазимим краищата посредством медни прочи(екранизиране), ше направим потенциалът на плетката еднакъв. Така външните смущения ще се поглъщат от плетката – STP(shielded twisted pair). При FTP една жица спираловидно обикаля около двойките и значително обезшумява сигнала. При коаксиалният кабел имаме медна жица обвита със диелектрик и плетка отгоре. При него високочестотното поле се затваря в диелектрика. LAN – кабелът са усукани двойки които директно предават сигнала.

Оптическите влакна се изграждат от специален прозрачен полимер. Основното им качество е пълното вътрешно отражение. Лъча като опре стената се отразява като следва и завоя. Проводниците имат радиус на огъване. Кабелът се състои от 1 или повече влакна в пластмасова обвивка. Влакната са еднопосочни, едно влакно за една посока.Сигналът не може да се подслушва и няма външни шумове. Светлинния лъч губи от енергията си за да преодолее радстоянието , така де получава основният проблем - затихването. Сигналът не може да се модулира честотно(ще промени дължината на вълната), затова се модулира амлитудно, като това най често става с мигане – пращане на импулси.На единя кай стои източникът който трябва да е кохерентен(с еднаква дължина на вълната) и безинертен – лазер. А но другия се слага специален светлодиод който преобразува светлината в ел. ток. Недостатък е че няма възможност за използване на междинни накрайнице, също така и по-трудното и скъпо възстановяване на кабела. Т.к. светлината е с висока честото скоростта на предаване на данните е максималната теоритически възможна – няколко Gbps. С оптически влакна се правят опорните мрежи – гръбнакът на големите лан мрежи или градските мрежи.

Въздушна технология – ефирът. До 10км над земята има въздух , а съществуващите радиовълни могат да се разпространяват и над това. Не се допуска използването на вълни след светлинния диапазон за предаване на информация. За пренасяне на информация се използва сателитната връзкал при ниея спътници се изкарват в орбитата. По високите имат по малко триене и по кръгова орбита, по ниските са с по елептична и след време падат- ако сателитите са над екратора се създава илюзията че те не се движат, гледани от земята – геостационарни спътници. Спътникът има две параболични антени и приема само сигнали с определена носеща честота и може да предаде сигнал както към земята така и към съседните му спътници. Плюс е че обхващат цялата земя, минус е големината и точността на земните антени както и цената на канала.

Infrared- работи в стая с малка мощност. При него може да има един хъб към който да се държат всички и в посредствие да се отдели отделен канал за всяка връзка.

5.Канално ниво – кадри, предаване, грешки, номерация, прозорци.

Каналното ниво има задачата да структуира битовете, които се предават към друг абонат, така че той да може да различи началото и края. Въвеждат се кадри, които са порция от битове която се приема за вярна в отсрещното канално ниво. Данните които се предават остават непроменени, кадрите се различават в служебната си част.

Каналът за предаване може да е point2point или multypoint. При P2P имаме от точка до точка, може да е симплексен, дуплексен или пулодуплексен. В канала се предават кадри от бетове, като е важно различаването на началото и края. От гледна точка на различаването на началото и края имаме байтов и битов поток. При байтово ориентираните канали имаме твърда структура от байтове които са натоварени със значение. Има запазена поредица за SOH (start of header), EOB(end of block) и тн. При предаване на даните ако се получи SOH кадърът се прецаква. За това се поддава служебна комбинация DLE – предавателят е длъжен да сканира кадъра. Той сканира всеки байт и ако установи че ще предава служ инф пуска едно DLE и след него информацията. Приемникът при сканиране като види DLE – го маха и не сканира следващия байт. Това обаче се оказва неефектимно заради 8 битовото DLE и постоянната проверка. Затова се измисля битовият поток. При него битовете се гледат един след дрег. Комбинацията 01111110 се определя за начало и край. При преход от 0 към 1 се формира брояч който брои 1-ците. Ако изброи 6-1ци и после не е 0 – грешка в кадъра, ако е 0 – броячът се нулира. За да не се среща нашата комбинация в данните при предаването ако изброи 5-1ци добавя 0 след тях и нулира брояча. При приемането ако след 5-1ци има 0- маха я и нулира брояча, ако са 6-1ци и 0 – то е дошъло начало/край на кадъра. Така с висока скорост се обработва маркера.

Много често след маркера имаме bit-stuffing синхронизация на каналните адаптери. Каналът трябва да е абсолютно надежден. Проблемите които могат да възникнат са - загуба на кадър, изкривяване на битове, прекъсване на кадъра и поява на дубликат. Затова А държи кадъра в буфера си докато не получи потвърждение от В за успешно получаване. Ако кадъра се загуби – В не знае това, затова А има timeout, ако то изтече преди потвр. А праща кадъра наново. Ако това се повтори няколко пъти А може да обяви канала за неработещ. При bit-styffing(синхронизацията) може да се получи грешка и В да изхвърли кадъра. За това в рамките на кадъра се слака контролна сума(CRC код). При четене на битовете се смята този цикличен код и накрая се сверява със записания в кадъра. Ако В разбере че кадъра е сбъркан може да си трае – тогава при изтичане на timeout-а А ще прати наново кадъра (недостатък – може да се обяви канала за невалиден, предимство- логиката на В не се усложнява) или да изпрати отрицателно потвърждение на А и да получи наново кадъра(предимство – отр потвр може да изпревари timeout-а). Ако потвърждението на В се загуби, А ще прати повторно кадъра и в В ще се получи дубликат. В трябва да разпознае че е получил копие, затова кадърът трябва да е уникален за някакъв период на време. Варянт е слагането на алтернативен служебен бит(1-0,2-1,3-0,4-1). Така като А получи потвърждение за кадър с бит-0 праща следващия с бит-1, ако не праща същия кадър. При В ако се получат 2 кадъра с еднакъв бит то тогава имаме копие. Този протокол се нарича „спри и чакай потвр”. Процедурата е неефективна поотношение на скоростта, имаме престой в канала. За да се ускори процеса се правят процедури с прозорци. Отварят се буфери с поече места за кадри. А предава на В серия то кадри. В буферите на А се слагат кадрите. И те се пращат на В в прозорец с големина 4, т.е пращат се първите 4 кадъра- имами спри и чакай целия прозорец. В ги проверява независимо и връща потвръждение с най-големия номер на кадър получен коректно. Ако 2 пропадне – се потвърждава кадър 1ллтогава прозореца се плъзга до там и подава нови 4 кадъра. Тук алтерниращ бит не върши работа затова се прави последователен номер на кадъра(цекличен кадър). Мин дължина на това поле трябва да е двапъти големината на прозореца за да се разпознае откъде е дошъл кадъра.

6. Управление в канала – HDLC формати

Към 70-те се прави стандартизация и се създапа процедура за управление на канала – HDLC(hign-level data link control). Формат на кадъра -... Полето за адрес се използва за многоточкови канали, при P2P полето остава празно. Полето CRC – контролна сума , се вкарва хардуерно в кадъра. Полето за контрол определя 3 типа кадри
-информационни- seq number – посл. номер на кадъра, next – прикрепено потвърждение в обратна посока, P/F poll-final (0/1), последният кадър от сесията е с бит F. когато имаме 2 адаптера вдигаме poll за да означим че насрещният адаптер не трябва да чака подходящ трафик за пращане на прикрепено потвърждение.
-Неприкрепените съобщения се наричат супервайзорни кадри (форсирано потвърждение). Тип: 0-receive ready-потвр с номер на следващия, 1-reject – от кадъра с номер next всички са отхвърлени, 2-receive not ready – получил съм всички кадри до next не искам да получавам повече, 3-selective reject – отхвърля само кадъра с номер в next.
-Неномерирани – в тях няма номер. Използват се за служебна настройка на каналните адаптери. Има 32 команди за работа с адапторите
DISC – disconnect – когато се иска прекратяване на трансфера
SNRM(set normal response mode)- А кара В да влезне в режимнормално отговаряне. Връзката е асинхронна и асиметрична
SABM(set asynhronic balamced mode) – за равнопоставяне на А и В
FRMR(frame reject) – използва се за указване че е получил синтактически верен кадър, но семантично не отговаря на мястото си.

7. Метод на достъп до съобщителната среда в ЕТЕРНЕТ.

ЕТЕРНЕТ е специален многоточков канал, в който няма детерминиран достъп до канала. При детерминирания достъп има предварително определяща дисциплина, която казва кой и кога може да праща. В коаксиалният кабел няма никакво специално устройство и си изработва метод на достъп CSMA/CD(carrier sense medium access/ collision detection)-метод на достъп със следене на връзката и разпознаване на конфликтите.

Сигналът се пуска в двоен манчестърски код ДМК, основното при този поредица е че в жилото се получапа постоянна токова съставка. Когато има нещо асинхронно спрямо 0 се получава различна токова съставка и така се разбира че има предаване.(това е следенето на носещата). Ако дакато един абонат предава сигнал, втори рише да се включи , тогава като се двете съберат двете импулсни поредици, общата постоянна токова съставка се качва толкова че не може да бъде изпратена от една точка само.

Коаксиалният кабел е така изчислен като диаметър, че в двата края като се сложат калибрирани съпротивления те да поглъщат полето на кабела. Когато една станция започне да предара сигнал, се предава електрическо поле, което пристигайки в краищата, ако те са просто отрязани, ще се отрази и ще предезвика заглъхване. Затова краищата се терминират с точно определено съпротивление, което до поглъща електромагнитното поле.

Кабелният сегмент се взима с такава дължина че предаването м/у две станции да става за не повече от 4.8ms(микро секунди). Когато една станция вижда че носещата е свободна тръгва да предава. След 4.8ms носещата ще се появи на другият край. Ако той обаче малко преди това време реши да предава, двете носещи ще се насложат и ще тръгнат към първият край. Наслагването ще стигне първият край след 4.8ms. И тогава той ще разбере че има колизия, т.е. тя може да се получи за макс 9.6ms. Затова преди да започне да предава една станция трябва да изчака 9.6ms, за да може свободата на носещата да се разпространи в целия канал. Колезията се хваща, т.к. два противоположно движещи се сигнала не могат да се заглушат. Една станция трябва да предава минимум толкова битове, че кадърът да заема целият канал, т.е. 9.6ms. Ако за това време не стане колизия, то каналът е стабилно зает. Когато колизията се усети, се праща сигнал JAM, който да не може да се сбърка с CRC(че кадърат е верен). Кадри с <64 бита се снае че са резултат на колизия.

Всички подслушнат канала и когато се появи нещо в него, те са длъжни да го следят, за да видят дали е предназначен за тях. Оригиналното съобщение минава през всички и те не могат да го променят. Кадърът се записва в буфер само при наличието на празно място. По изчезването на носещата се познава че кадърат свършва. Ако кадъра е по малък от 64 бита директно се изхвърля. Първите 48 бита от кадъра се сравняват с МАС адреса. Ако е само от 1ци то това е broadcast и е за всички. След което се проверява и контролната сума дали съвпада. И чак тогава кадърат се приема. При приет кадър имаме прекъсване и активеране на драйвера, който трябва да дръпне кадъра от буфера на адаптера. И това трябва да стане максимално бързо.

Етернет е за малки и средни натоварваня, т.к. при по големи броят на колизиите расте експоненциално.

8. Управление на канала ЕТЕРНЕТ. Превключватели и мостове.

Формат на кадъра в Етернет
-Preamble - :В за синхронизация на приемащата станция.
-Destination Address
-Source address. Адресите трябва да са абсолютно уникални, като веднъж използван адрес не може да бъде даден пак.
-Length – минемалният размер на данните е свързан с обирането на колизиите.

Подформат на кадър е IEEE 802.2. При него след полето зо дължена има 1В Destination Service Access Point, 1В Source SAP, и 1 контролен байт. SAP отговарят на портовете. А типът показва за какво ниво е.

Друг формат е Етернет2 – стандартният + тип вместо дължина

Ако данните са <64В се добавя padding.

IPX протоколната система е създадена за Етернет, но след 95та е изместена от TCP/IP. Там се е използвал Етернет2. А в съвременният кадър отпада и полето length.

HUB-ът е повторител който разпространява кадрите към всички активне станции. Ако стане колезея пак я праща на всички. HUB-а се превръща в switch като му се добави зарпознаване на адреси. Станцията заявява своя етернет адрес и switch-а си строи тоблица с адресите, свързани с даден негов порт. Switch-ът е маршрутизатор с комутация на каналите. Като влезне кадър в switch-а, DA се сравнява с таблецата на активните станции, намира порта към който е закачена и превключва на връзка 1-1. Същевременно изхода на получателя може да се свърже със друга станция 1-1. По този начин switch-а работе в пълен дуплекс(хъба е полудуплекс). За да може switch-а да се справя със скоростта трябва кабелите да са добре екранирани. Switch-а намалява броя на колизиите но не ги премахва.

Switch –овете могат да са 2 вида – без буфериране в switch-а и с буфериране.

Те магат да се свързват един с друг като – daisy chain(където няколко switch-а работят във верига. Само един от портовето на switch-а е подходящ за свързване с другия) или в стек(правят един общ стек, свързани са с дебел кабел с много изводи)

HUB-овете се свързват във верига и всички сигнали ги обикалят.

Bridge-а свързва 2 самостаятелни канала. Работата му е чисто селективна м/у двата трафика. Той има по една таблица и за двата сегмента, които свързва и препредава ако види че трябва да се предаде кадър от единя сегмент към другия.

-source routing – кадъра се засилва към bridge-а. Кадърът се допълва с верига от етернет адреси за bridge-овете, през които трябва да мине.

- spanning tree – няма вериге от bridge-ове, а те трябва сами са съобразят накъде да изпратят кадъра.

Преимуществото е, че колизиите се запазват в границите на сегментите. Bridge-а играе роля на физически разделител.

Repeater – повтаря от единя сегмент в другия. Вече не се ползва.

9. Маршрутни алгоритми – обхождане, наводняване, метод на Берън

При мрежите с маршрутизация хостовете са свързани към специални възли, които осъществяват разпознаването на входящия поток и доставят на съответния получател само определените за него данни.

1. кадърат влиза в каналният адаптер
2. буферера се
3. декадрира се
4. работи с пакета
5. по информацията в пакета определя накъде да отиде
6. слага се в опашката на даден канален адаптер
7. кадрира се

Маршрутизацията може да стане чрез датаграми(с тях работи switch-а) или с пакетен виртуален канал(с тях работи rooter-а).

Маршрутизация с наводняване. Това е най простата стратегия. Когато пакет влезне от даден канал, в рутерът той се размножава и слага по едно копие във всички канали. Така ако всички възли работят по този начин със сигурност пакетът ще стигне до където трябва при това пътя ще е най късият възможен.

Маршрутизация с обхождане. При нея когато попадне пакет във възела той се пуска по случайно избран канал.

Метод на Берън. (метод на горещия картоф) при него като влезне пакет, рутерът га вкарва в най-късата изходяща опашка. Това е изолирана-адаптивна маршрутизация.

10. Статична и централизирана маршрутизация

Статична маршрутизация. При нея във всеки рутер ръчно се въвеждат маршрутизиращите таблици. Нека в даден възел постъпи пакет за D. В таблицата имаме H-0.5, A-0.25, I-0.25. Генерира се случайно число от 0.00 до 0.99, ако числото е от 0-0.5 праща се към H, м/у 0.5-0.75 A, м/у 0.75.0.99 I. A от там съответния рутер поема маршрутизацията. Имаме 3 маршрута защото алгоритъмът не е адаптивен. Самата таблица не се променя често и стои доста време във възела. Тя не отчита натовареността или промените в пътищата, затова ни трябват резервни пътища.

Централизирана маршрутизация. Тук един от възлите се прави на център за маршрутизацията и периодично набавя информация за мрежата. На определено време се пускат служебни пакети които си пробиват път до централния възел като събират информация за мрежата. За всеки възел се строи маршрутизираща таблица, която отчита динамичните промени. Таблиците се стоят и се изпращат към съответните им рутери. Таблиците са служебни пакети само че със посочен получател. Предимства – таблиците са добре обмислени и осъвременявани. Проблеми- Ако нещо се случи с маршрутизатора цялата мрежа е в аварийна сатуация. Този проблем се решава с backup маршрутизатор. Проблем е и когато мрежите са големи. Тогава при изпращането на маршр таблици от централния маршрутизатор те ще стигнат със закъснение при до далечните, така синхронизацията с получаването на новите таблици няма да е едновременна и може да се получи объркване.

11. Разпределена маршрутизация с дистантен вектор

При този алгоритъм всеки възел сам си смята маршрутизиращите таблици на определено време.

1. Всеки рутер периодично измерва разстоянието/натоварването до съседите си като праща echo пакети. В тях записва времето на поставянето на пакета в изходящата опашка. От другата страна при приемане на такъв пакет възелът веднага го връща като записва своето време на изпращане. По този начин всеки възел съдържа вектор на разстоянията до всички възли в мрежата. Периодично възлите си обменят тези вектори за да могат те да се обновят.
2. По простият варянт е да се смятат броят на рутерите които свързват възлите- т.е. броя на хоповете. В началото А не работи и разстоянието се счита за безкрайност. След като се включи праща ехо пакет и като В му отговори значи чи В е на 1 хоп. И таблиците се разпространяват. Ако обаче А престане да работи това ще се разбере много бавно. Подобрения – за безкрайност се използва числото 16. Т.е. разстоянието м/у два възела не може да е повече от 15. ако оптималният път се влоши, новата стойност на оптималният път се слага в съседа. Предимства – трафикът не натоварва мрежата защото служебните пакети се разменят само със съседите. Ефективен е при малки мрежи.

12. Маршрутизация със следене състоянието на връзката

Link State Routing е алгоритъм със следене на връзката. Основните идеи са

1. Всеки рутер установява кои са му съседите и научава мрежовите им адреси. При включването рутерът изпраща hello пакет по всички изходящи линии. В този пакет се представя и пита за чуждият идинтификатор. Рутерити предварително трябва да имат уникални идентификатори.
2. Измерва се разстоянието до всеки от съседите. Това става с ехо пакети, като най-често (отиване + връщане)/2
3. Констуира Link State пакет за себе си, в който се съдържа неговият адрес и информация за съседите.
4. Изпраща се формирания LS пакет на всички рутери. Чрез метода на наводняването мрежата ще се натовари. Освен това LS пакетите периодично се обновяват и пак трябва да се наводнява. За да се избегне това се въвежда поредният номер на пакета и той се пази в рутера. Така рутерът не праща пакета в посоката от която е дошъл. Ако няма номера на този пакет го препраща, ако го е записал вече направо унищожава пакета. Ако дойде пакет от даден възел с по малък пореден номер от актуалният за възела, то той се пренебрегва, а ако е по голям се заменя. LS пакетите се пращат периодично или при промяна.

Недостатъци – при нарастване на възлите – LS пакетите се увеличават. Също така ако рутерът угасне брояча на пакетите ще се нулира и така пакетите автоматично ще се отхвърлят. За избягването на това се слага AGE на пакета, като всеки възел я увеличава с времето за престоя си. Така ако даден възел има текущ пакет с номер 25 и получи пакет с номер 1 но с по-голяма възраст , ще го върне. Но ако възрастта е по-малка най-вероятно това е нов пакет.

5. Всеки рутер събира множество пакети и строи граф на мрежата за себе си по митода на дейкстра- .граф на най-късите пътища.

13. Йерархична Маршрутизация

При нея имаме разделяне на области. Като при повече рутери имаме повече нива.
При Link State алгоритъма трябва да се знае кои възли са в дадената област и наводняването става само в нея. Така алгоритъма се усложнява но работата се подобрява.
Йерархията води до намаляване на маршрутизиращите таблици за сметка на неоптимални пътища.

14. Засръстваня и управление на потоците в мрежата

При мрежите с маршрутизация хостовете са свързани към специални възли, които осъществяват разпознаването на входящия поток и доставят на съответния получател само определените за него данни.

Ако възелът има 3 канала от които идват данни а през един да излизат пакети, може да се получи опашка на изхода. Когато свършат броя на свободните буфери възелът си затваря входовете. Постепенно изпращачите, чакайки го да отвори буфер, също ще задръстят и така се разпространява задръстването.

Стратегии при задръстваня.

1. Предварително регулиране на буферите с цел избягване на задръстванята. При входната част на всеки канал задължително има един свободен буфер, за да може да се разгреждат идващите пакети – за къде са. Ако пакета трябва да лъде запазен тогава се освобождава някой от изходните буфери и пакета се слага там.

Ако възлела знае какве са потоците минаващи през него би могъл да разпредели буферите си да рабътят спрямо тях. Когато служебният пакет който прави съединението мине през възела, се запазват 4 свободни буфера които да работят само с това съединение. Ако всички буфери са били запазени тагава този пакет се връща.

При ползване на съединението се пращат 4 пакета. Възелът ги получава един по един и го слага в заделените буфери. Чак когато възелът изпрати и 4-тият пакет, праща потвръждение на предният възел. Чак тогава се праща нова порция от пакети.

Недостатък е това, че буферите стоят неизползвани, ако съединенията не работят. А и немогат да се дабавят нови съед.

Това е единствената стратегия която гарантира виртуален канал без задръстваня.

2. Отстраняване на пакети при задръстване на рутера. Пакетите не се отстраняват произволно, преди това се разглеждат. Пакети с прикрепено потвръждение не се отстраняват. Аерленд предлага ограничение в/у броя на прикрепените буфери към изходните опашки. m-макс брой да ления, k – брой свободни буфери, s – брой изходни линии. M=k – лошият случай, m=k/s –възелът няма да се задръсти, но няма да работи ефективно, m=k/sqrt(s) – най- доброто, при него изходната опашка набъбва до този размер, след това пакетите се изхвърлят за да не нарушат работата на опашкато. Ако опашката е пълна и дойде пакет за нея, тя се проверява за потвърждение, ако няма такова пакетът се изхвърля. Ако обаче има потвърждение – изважда го и го обработва, може да се наложи да си освободи буфер заради него. При по къси опашки задръстванята са по-малко, но се отстраняват повече пакети.

3. Ограничаване броя на пакетите, които влизат в помпютърната подмрежа. В мрежата се пускат да циркулират определен брой служебни пакети, представляващи разрешения. Когато абонат иска да прати пакет рутерът прихваща разрешение и на негово място праща пакета, получен от абоната. Когато на другия край рутер прати съобщението на крайния абонат, се генерира ново разрешение.

Подходящо е за датаграми, където няма виртуални канали. Недостатъците са че има проблем при загубата на маркерите. Този подход решава зодръстванята на глобално ниво, затова алгоритъма трябва да се усложни – да има различни разрешения за отделните области.

4. Управление на потока. Потокът се разглежда като информационен поток от пакети, които даден абонат вкарва в мрежата до друг абонат. При достатъчно голям поток от единя абонат, другият може да не се справи и да се получи задръстване. А ако още няколко абоната пратят голям поток воже да се получи задръстване в ммрежата. Затова самата мрежа трябва да казва на транспортното ниво, че се претоварва и то да намали ширината на прозобците.

5. Регулиране на входа при задръстване. Възелът има определен брой свободни буфера. Когато се заемат 80% от тях, възелът праща предупреждение в потвържденията си. Предният възел вижда от къде пристигат тези пакети и ти разпраща в обратната посока. Така се стига до източникът, и крайните рутери лесно ограничават входа си от този абонат. Когато 80% намалеят, възелът праща съобщение че се е поосвободил.

Ако системата попадне в deadlock възлите си мислят че става дума за задръствяне и чакат да се освободи. Ако един от възлите разбере че е в такова положение и изтрие един пакет, нещата ще се оправят. Това обаче е малко вероятно да стане.

15. Мрежов протокол IP – адресация, подмрежи и маски.

За да позволи скелетната мрежа съвместна работа на различни мрежи то тя трябва да позволи работа с пакети с различни дължини. Затова е създаден IP протоколът, за да пренася пакети на различни мрежи. Работи с датаграми с размер 1500B, т.к масовата автономно система е етернет, а там накетът е с такава големина. Понастоящем това е най-разпространеният протокол на мрежовото ниво.
Header-a на ip-пакета не е с фиксирана дължина. Може да е макс 64байта, като 20 от тях са задължителни.
Version – версията на IP протокола
Type of service – указва дали става дума за файлов трансфер или интеративна заявка.
Total length – общата дължина на пакета в байтове
Identification –източникът на датаграмите слага стойност с която да се идентифицера че тези пакети са от него.
DF- don’t fragment. Минималният размер на максимално допустимата стойност на фрагмента е 576B. Един пакет от 500B с пуснат DF не трябва да се фрагментира. Ако е 1500B и е пуснато DF, то може да се спази , а може и да не се спази.
MF – more fragments – казва че следват още фрагменти. Като размерът на фрагментите трябва да е кратен на 8.
Fragment Offset – показва къде в дейтаграма първоначално започва фрагментът
TTL – time to live. Времето на живот на пакета. В секунди или в брой хопове. Допуска се един рутер да увеличи ttl-то ако пакетът се забави дълго време в него.
Protocol - указва кой е транспортният протокол който се ползва – TCP/UDP
Header Checksum – контролна сума върху header-a
Source address – адрес на източникът
Destination address – адрес на палучателят
Options – полета кратни на 4B.- Security ; strict source routing – казва точно маршрута от който да мине пакетът; loose source rooting – списък на рутери през които е добре да мине пакетът; timestamp – при движението на датаграмите през рутерите, те си записват тук id-то и точното време на преминеване; record routing – запесва IP адресите на рутерите, през които се минава IP адресите се дават от специална организация Network Identification Center. И тези адреси са уникални. Фиксирани адреси – 0.0.0.0 нашият хост, 127.0.0.0 lapback, 255.255.255.255 broadcast.
Subnet – става с мрежовата маска. Тя се състои от поредица от 1ци последвана от 0ли. Когато входният буфер получи адрес, той знае че става дума за подмрежи. Наслагва маската и намира за коя подмрежа и кой хост става дума.
IP адресът е уникален. Всеки хост видим в пространството трябва да има уникален адрес. Този адрес не е уникален сам за себе си. Той не може да се върже завинаги за даден адрес.

16. Преобразуване на IP адреси и физически адреси.

Всеки хост се идентифицира с IP и MAC адрес. Но формула за преобразуването от единя към другия няма.

Протоколът ARP се използва за намиране на MAC адреса. IP-то се поддава на ARP и се пита дали го има в мрежата и какъв е съответният му MAC адрес.

Всеки хост има в себе си ARP cache, който е празен при включването му. Когато хостът научи за някакво съответствие м/у MAC и IP, си го записва в кеша. При заявка, ако има информация я връща. Ако няма се пита мрежата с ARP запитващ служебен кадър към всички. Във Source Address-а слага своя адрес, после IP адреса който търси. Всеки който получи такава заявка първо записва съответните на източника IP и Mac адрес, а после проверява дали неговото IP е търсеното, като праща пакета нагоре по TCP/IP протокола. Отговорът се връща пак като broadcast, за да може и останалите да си запешат данните в кеша.

Проблем е че в основата на ARP има broadcast питане. В етернет не се знае докъде ще стигне този broadcast. Може да излезне извън рамките на текущият сегмент, например през bridge. Винаги трябва да се следи предаването по broadcast. Ако bridge-а не пропуска broadcast той ще насочи трафика към default gateway и така питането ще излезне навън и връщането му ще е по трудно.

Обратният протокол на ARP е RARP(reverse arp). При него по MAC адрес се пита за съответния IP адрес. Ако някой веде в таблицата си запис за MAC адреса, връща отговор. RARP също разчита на broadcast. Ако IP не се пази в паметта, трябва да имаме RARP сървъри, които за пазят съответствията.

17. Маршрутни протоколи RIP и BGP.

Основното при интернет е наличието на автономни системи- самостоятелни мрежи със собствена организация. Казва се че в АС маршрутизаята се обслужва от Interior Gateway Protocols а извън тях от Exterior GP.

Първият IGP протокол, който се вкарва в интернет е **RIP**. Базира се на лекия алгоритъм за дестантния вектор. А EGP протоколът който се използва е Border GP.

Особеност на RIP протоколите е, че те се зараждат в началото на интернет и нямат много изискваня. Така се създават много RIP протоколи и не се знае как точно правят нещата рутерите.

Формат на RIP1

COMM – командното поле показва дали датаграмата е заявка или отговор. Заявката изисква ругерът да изпрати цялата или част от маршр. си таблица.
Ver – версията на RIP протокола с която работи съответният рутер
AFI(address family identifier) – показва кое семейство от адреси се използва.
IP – ip на рутеря вътре в мрежата
Metric – броят на хоповете до рутера. От 1-16, при 16 се счита за недостижим

Формат на RIP2

Route Tag – използва се за да се различат вътрешни от външни рутери. С него се номерират някакви гранични точки на АС, които са свързани с EGP рутери.
RDom – номер на рутинг процеса. Това обаче води до усложняване на алгоритъма и CISCO не го приемат.
SN – с него относно IP се определя мрежовата и хост частта.
В RIP2 се добавя и поле за аутентикация на рутинг пакет. Създават се полета за автентикация на датаграмата. Първо се указва кой говори и второ че има правото да говори(ауторизация).
За **BGP** протокола автономната система е една точка в мрежата. Основното при него е ръчната намеса при определяне на поритиката на маршрутизация. Поради това в BGP имаме различни категории като
-stub network – мрежа с вход към само един BGP рутер
-multiconnection network – повече от един BGP рутер по границата.
- транзитна мрежа – свързва 2 BGP рутера
Смята се че BGP рутерите са обозримо количество. Прието е да се използва алгоритъмът на distance вектора с описание на пътища.

18. Маршрутни протоколи OSPF. Безкласова маршрутизация.

OSPF е маршрутен протокол, който се базира на link-state алгоритъма. Тук е задължително протоколът да е отворен и алгоритмите трябва да се знаят. Автономната система се разбева на области. При съседни рутери в ЛАН мрежа няма смисъл всички те да си предават таблиците, затова единя рутер се взима за designated. На него прилежащи стават рутери извън локалната мрежа.
Формат на OSPF
Ver – версията на протокола
Type
- hello – определят се топологично съседните рутери и destination рутера
- database description – определяне на прилежащите рутери (след hello)
- Link-state request – изисква обмен на маршретните таблици
- link-state update – (основно такива пакети се обменят)
- Link-state acknowledgement -
Length – дължина на целия пакет
RouterID – идентификатор на рутера който издава пакета
AreaID – идентификатора на областта за която се издава пакетът
Checksum – контрална сума
AuthType
Authentication – дали този пакет е издаден от този рутер
Data – съдържа данни свързани с маршрутизацията

19. Транспортно ниво – процедури за съединенията.

Задачата на транспортното ниво е да предаде едно произволно дълго съобщение с помоща но трето ниво. Може преди да започне предаването на съобщението да се установи контакт и двата абоната да се разберат за конкретни неща. Транпортът на съобщението трябва да дава някаква надеждност, независимо каква е надеждността на мрежата. От едно съощение се правят серия от пакети, които се поддават на мрежорото ниво. На тях се слага отделна номерация, както и отделна контролна сума за цялото съобщение.

Установяване на транспортно съединение.

Three-way handshake. А праща заявка за установяване на съединение с максимален размер на съобщенията 4MB и започва до ги номерира от 1. А ще чака потвърждение от В и чак след това ще отвори буферите си. Ако В отвори свойте буфери при изпращане на потвърждението и то се загуби, А ще се откаже а в В ще има неизползваеми буфери и ще чака да дойдат пякети. Затова А трябва да прати потвърждение на потвърждението и чак тогава В да отвори буферите си. Ако второто потвърждение се изгуби, А ще почне да праща пакети и В ще се усети че става дума за изгубено потвърждение. В обаче няма да има заделени ресурси.

Транспортните съединения са симетрични – т.е. и двете страни могат да разпаднат съединението. А трябва да прати заявка за прекратяване на В и В да я потвърди. При нормална работа трябва да мине цялото съобщение, единя да потвърди предаването му и да поиска край и другият когато приключи работата си да потвърде че и той е готов и също иска да разпадне съединението. Алгоритъмът за разпадане не е един и същ – може моментът на изпращане да не е подходящ и В директно да откаже, или да каже че ще прекъсне съединението но след малко.

20. Транспортни протоколи TCP и UDP.

Данните на **TCP**(transmission conrol protocol) се носят от IP дейтаграмите. TCP протоколът е със съединение, и осигурява надежден транспорт на базата на назежден IP транспорт. Поддържа разделянето на сегменти, прехвърлянето им и подрежданито им от другата страна. Има възможност обменът м/у два абоната да се раздели на отделни потоци.
IP адреса + № port дава идентификация на входната точка на потока, който ще се обработва. Портовете са с номера от 0-65535, като от 0-1023 са well-known. Това са портове но които чакат за заявка определени сървъри(80- web server).

TCP пакет

Source Port
Destination Port
Sequential Number – поредният номер на сегмента от началото на потока.
Acknowledgement number – потвърждение до къде е получил сегментите. Следващият номер след последно получения.
Data offset – размер на хедъра в 4B думи
Флагове

- Urgent – спешни данни за пращане
- Ack – придружава валиден номер на потвърждение
- Psh – означава да се прати push команда в другия край. На всеки край има буфер където се трупат пакетите, при получаване на push команда приемателя изпразва буфера.
- Rst – reset – изчистване на броячите на връзката, на изпращача и на приемника
- Syn – synchronized – изпращачът иска да синхронизира броячите.
- Fin – няма да се изпращат повече данни

Window – размера на прозореца за изпращане в октети
CheckSum – контрална сума в/у хедъра и данните
Urgent Pointer – показва от къде в данните почват спешните данни
Options – опции като максимален размер на сегмента и т.н.
Padding – затваря опциите

UDP(unreliable datagram protocol). Единственото което прави е да добави портовете и контролната сума

21. DNS система за именоване. Процес на резолвинг на имената.

DNS- domain based name service върви заедно с IP адресацията. Всяка операция в интернет се свежда до знаене на IP адреса на предоставящия услугата. Но по двоичен адрес не може да се разбере разположението целта и т.н, затова се създават домейн имената.
Домайните от първо ниво са ограничени и вкарването на ново е трудно. Целта е да се разделят сервизните точки но високо ниво. В началота сащ си дават собствени разделения – com(commercial),edu(education) и т.н. П воследствие се оказва по довра идеята за разделение по държави т.к.така йерархията носи някакава яснота- bg,ru,uk...
URL – uniform resource location www.fmi.uni-sofia.bg (име на хоста . домейн от 3то ниво . 2ро ниво . 1во ниво). Зад всяко URL стои реален IP адрес. Човек може да си купи име на домейн от 2-ро ниво
FQDN – fully qualified domain name – е общо взето като URL, но може да е по подробно, като зададе порт или друга част на компютъра.
DNS се появява с цел да се избегне съхраняването и разпознаването на URL-то на място. Техническата част е да се реализира разпознаването на URL.
Заявките към DNS сървъра са рекурсивни, итеративни и инверсни. Започва се с рекурсивна заявка. DNS сървъра има именно пространство – кешово. Ако там адреса го няма , в зависимост от конфигурацията заявката се предава рекурсивно или итеративно.
Рекурсивната заявка отива към firewall-a и после към remote DNS. RootNS не е само един, а ся много разпръснати по света. RDNS таблиците имат еднакви леви страни, но IP адресите мотаг да са различни за домейните от първо ниво. (root NS се пита за bg, после bg NS се пита за uni-sofia, после uni-sofia NS се пита за fmi , и после за IP-то на съответния хост). По пътя на рекурсивните заявки от Remote DNS резултата се връща към local DNS и той го връща на резоувъръ. На базата на полученото IP се прави TCP съединение с web-server-a на порт 80.
Инверсните заявки търсят но IP адрес съответният FQDN. Използван се във firewall-ите.
На DNS resolver-a се подава URL и той връща IP адреса. Резолвъра обикновенно е част от TCP/IP протоколът, но може и да е отделен модул който да се вика самостоятелно. Процесът на преобразуване минава през следните етапи:

1. проверява дали URL-то не е името на локалният компютър
2. проверява в DNS кеша на компютъра(там се съхраняват последно използваните URL-та)
3. проверява в hosts файла.
4. пита локалният DNS сървър. Всеки компютър има различни начени да пази IP-то на своя DNS сървър – с явно указване или зрез broadcast питане. Праща се заявка към DNS сървъра с търсеното URL. И от там се връща IP адреса.

В Unix процеса спира до тук. Ако не се върне IP адрес значи той не съществува.
При Microsoft

5. търси в директорията WINS
6. търси с локален broadcast в LAN
7. търси във файла lmhosts като име на машина.

Зоновия файл съдържа записи от различен тип, но основно са име-> IP адрес. PDNS управлява своя зона и има поне един такъв зонов файл, който се съхранява в/у диска и може да бъде изменян от него. SDNS също има зонови файлове, но той само ги получава и ги ползва. Зонов трасфер се налага при всяка промяна на зоновия файл. Задачата на SDNS е да облекчи работата и трафика към PDNS, като е задължително всеки PDNS да има поне един SDNS сървър.
Master DNS сървъра има задачята да разпространява зоновите файлове от PDNS към SDNS. Като не всеки SDNS получава зонов файл пряко от PDNS.
Когато се регистрира домейн първо се проверява дали той не е регистреран вече. Когато домейнът се създаде ни трябва физически DNS сървър, който до създаде зоновия файл за домейна. След което за разпространениета на зоновия файл трябват поне 24часа.

22. Файлов трансвер в Интернет.

FTP е протокол зо предаване на файлове в рамките на TCP/IP. Това е протокол на сесйно ниво м/у два ftp-клиента.

Имаме FTP сървър, който се състои от DTP и PI(protocol interpreter). PI е закачен за порт 20, а DTP на 21. В клиентската част имаме user interface (UI) мадул, който управлява начина на поискване на услугата. UI модула оформя URL, той минава през резолвинг и се връща IP адрес. IP адреса се подава на PI със задачата той да установи TCP съединение м/у PI модулите , по това съединение се обменят команди и отговори. По него се подава името на желания файл. И на транспортно ниво се създава второ ниво, независимо от първото по което се движи файлът. Цялата тази работа затваря една сесия. В рамките на една сесия могат да се обменят много команди и файлове.

Команди:

1. за контрол на достъпа – open, pass, user. Контрола зависи от настройките на ftp-сървъра
2. за трансфера – променя параметрите по подразбиране – Get, Put на файл.
3. за управление на файлове и директории – rmdir, mkdir, dir ако има права за това.
4. за помощ и състояния
5. за отговорите на сървъра- 1 положителен подготвитерин; 2 положителин; 3 положителен незабавен; 4 отрицателен временен; 5 отрицателен постоянен

Сигурност при ftp-трансфера - readonly за повечето директории - възможност за промяна на определени директории за даден юзер - криптиране на паролата

23. Електронна поща в интернет.

Електронната поща са криентси приложения като Outlook express и т.н. Те работят с протоколите на приложното ниво и имат user интерфейс.
SMTP – протокол на приложно ниво

- клиентът трябва предварително да знае FQDN на SMTP сървъра.
- Праща заявка на порт 25 за да провери дали е там
- Сървърът отговаря с 220 – Ready
- Установява се TCP съединения и сесия.
- Сред това клиентът праща Hello
- Сървърът отговаря с 250 – OK
- Клиентът праща азресът на изпращача
- Сървърът отговаря с 250 – OK
- На стъпка RECU TO се изпраща информация необходима за изпращането на писмото , за всеки абонат за който то е предназначено.
- Сървърът отговаря с 250 – OK , ако не отговори значи има проблем с някой от получателите
- Сред което се пращат данните за писмото
- За прекратяване клиентът праща QUIT
- И сървърът отгаваря с 221

Тялото на съобщението трябва да е в 7 битов ASCII код. Зада се избегне това преобразуване в протокола то се прави автоматично от приложението

- BASE 64 encoding – това което ще се кодира се разделя на октети, те се взимат на групи по 3, всяка група се разделя на 4 парчета от по 6 бита. Съдържанието на тези 6 бита се кодира със 7битов ASCII код.
- UUEncode – от Unix
- MIME encoding – за предаване на всякакви данни, където данните вървят с описатеи. Това е новият стандарт за пренасяне на данни, за писма със видео, аудио и др елементи в интернет.

SMTP протоколът се използва за изпращане на писма, но за да си получи писмото получателят трябва да е активен. За получаване се използва POP3 протоколът. Писмото се праща в SMTP сървър, а POP3 го прочита или изтегля от там.
POP3 юзера си има име, което най-често е часта преди @ в email адреса. Сесията се състои от поредица от команди и отговори- USER(user email), PASS(password), QUIT, START, LISI (приблизително големината на съобщението), TOP(показва горните редове на съобщението). POP3 сървърът може да се настрои да изтрива или препраща писмата при изтегляне и др.
При POP3 протоколът получателят не е свързан директно онлайн с подателя. Затова се прави едно голямо съобщение което се движи м/у сървърите като логическо цяло.

24. Хипертекстов протокол в интернет.

На web-сървърът се праща FQDN и той връща страница която браузърът изобразява. В браузъра има вградени правила за специално изобразяване на информацията.
Първата идея е, че при клиента има програма, която изобразява данни. Зад данните стоят и скрити неща, като например интернет адреси към други страници. Натискайки такава специална част програмата изтегля информацията от скритата част и се свързва към друг сървър. Така може да се създаде едно огромно разклонено дърво от препратки към web сървъри. Тези връзки се наричат хипервръзки, а текстове с хипервръзки – хипертекстове. Хипервръзката се състои от означена/разпознавателна част в текста и скрита част с пълният адрес на страницата за която се однася. Хипертекстовите страници са наредени в уеб-сървърите в каталози, като файлове с път към тях. И те трябва да се четът от специарни програми.
HTTP-hypertext transfer protocol – протокол в/у TCP. HTTP сървърът е програма, която разпознава HTTP команди, намира търсените страници и ги връща на HTTPклиентът. Най разпространеният HTTP сървър е Apache.
На браузъра се дава URL. Той прави резолвинг по FQDN и намира IP адреса. След това се прави съединение със сървъра на порт 80. после браузъра издава команда GET + местоположението на страницата на сървъра. HTTP сървъра намира страницата и я връща по съединението. При първите версии на HTTP, съединението се прекратява и страницата се изабразява.
HTTP proxy сървъра съдържа копия на често използвани страници. Така ако версията на страницата не е много стара я придвижва по бързо до браузъра, иначе ще предаде заявката на сървъра. HTTP сървъра не знае колко проксита има и затова те сами трябва де си обновяват информацията. В браузърите име възможност да се забрани минаването през проксито.
Команди в HTTP протокола с коита се обработва една страница

- GET – http request за страницата
- HEAD – http request за заглавието на страницата
- PUT – запазва страницата, това се разрешава след аутентикация
- POST - добавяне към именован ресърс
- DELETE – идтреване на страницата
- LINK – свързват се именовани ресурси
- UNLINK

В по новите версии на протокола може да има и други заявки към сървъра и сесията и съединението да не се затварят. HTTP 1.1 имат второ транспортно съединение, където се нахвърлят команди към сървъра преди да са се получиле отговори. Появява се и кеширането, което се използва най вече при гледането назад. В 1.1 има и header на host-а, което пазволява на един IP да има няколко web- сървъра. Има и метод trace за разпознаване откъде е минала заявката.
Заявкети минават в явен вид и могат да подлежат на атаки. Затова се разработва HTTPS протоколът. В него се установява сисея с асиметрични крючове и се обменя секретен ключ в рамките на сесията. Това е така нареченото ниво SSL – security socket layer. От гледна точка на трансфера по мрежата нещата са същите, само има няколко допълнителни стъпки за установяване на крюча.