Мрежата

3.1.Свещените протоколи

Интернет (Мрежата) съществува най-вече заради наличието и спазването на ясни и добре описани правила за обмен и обработка на информация между различни хардуерни устройства, компютри и компютърни програми. Тези правила се наричат протоколи. Те са детайлно описани в документи, наречени RFC (Requests For Comment). Можете да ги намерите лесно в Мрежата (просто попитайте любимата си търсачка за RFC). Вероятно имате най-важните и в поддиректорията /usr/doc/rfc на вашата Linux дистрибуция.

За нуждите на мрежовата работа има създаден т.нар. OSI седем-слоен модел, чрез който най-грубо работите в мрежата се разделят на слоеве (нива). За да съществува възможност за комуникация, протоколите поддържат функциите на няколко слоя едновременно. Модела изглежда така:

- Application layer предоставя големи приложенията, използващи услуги от по-долния слой. Пример за програма от това ниво е Netscape.
- Presentation layer осигурява конкретни услуги. Много протоколи дефинират такива услуги - FTP, HTTP, SMTP, POP3, DNS, BOOTP, DHCP, IRC и пр.
- Session layer организира правилния обмен на данни между програмите. Протоколите отнасящи се до това ниво (TCP, UDP) използват механизма на портовете, за да разпределят пакетите между различните програми, а също и различни техники за осигуряване на непрекъснат и коректен поток от данни.
- Transport layer пренася пакети информация от една крайна точка до друга в нехомогенна среда (между отдалечени устройства). Най-важният протокол свързан с този слой е IP.
- Network layer пренася пакети информация в хомогенна среда (между пряко свързани устройства), като например point-to-point връзка (PPP,SLIP).
- Link layer установява и контролира връзката между устройства, а също реализира схеми за компресиране на предаваните данни. Важни протоколи - LCP, PPP, ARP.

- Psychical layer кодира цифровите сигнали по аналогови линии.

Всички протоколи имат своето значение за работата на мрежата, и без съгласуваното им спазване работата на Мрежата е немислима. Ето по няколко думи за най-важните:

IP (Internet Protocol) пренася малки пакети информация между произволни два компютъра, закачени в Мрежата (без гаранция за успех на отделната пратка). Вижте по-подробно обяснение в статия 3.2 (следващата по ред)

TCP (Transmission Control Protocol) организира непрекъснат и подреден поток от информация между две програми (разговор между програмите). Използва като основа за работата си протокола IP. TCP изгражда връзката, накъсва разговора на пакети, изпраща ги и от отсрещната страна на връзката сглобява пакетите. Грижи се за повторно изпращане на загубените пакети и за пренареждане на тези, пристигнали в разбъркан порядък. За програмите TCP-връзката изглежда като два последователни файла - един за писане и един за четене.

UDP (Unreliable Datagram Protocol) за разлика от TCP при този протокол данните се изпращат без да се чака потвърждение, че са получени. Използва се там, където е по-важно комуникиращите програми да не се забавят поради чакане на потвърждението.

PPP

FTP (File Transfer Protocol) се занимава с пренасяне на файлове между отдалечени компютри. Един от първите приложни протоколи.

HTTP (HyperText Transfer Protocol) управлява обмена на WEB-страници и свързани с тях файлове и услуги. Млад приложен протокол.

SMTP (Simple Mail Transfer Protocol) извършва обмена на елекронната поща между отдалечени компютри (пощенски сървери). Пощенските клиенти го ползват при изпращане на писма.

POP3 (Post Office Protocol 3) чете електронната поща от отдалечен пощенски сървер.

DNS (Domain Name System) е удобна система за именуване на компютрите в Мрежата. Тази йерархична система от символни имена прави възможно по-лесното запомняне на имената на важните места, а също осигурява независимост на сърверите в Мрежата от доставчиците на трафик, които определят конкретния IP-адрес.

BOOTP, DHCP са два близки по семантика протокола за настройка по време на boot (началното зареждане) на IP-адрес и други мрежови параметри на компютри, които нямат предварително определени адреси.

IRC

В Linux голяма част от реализциите на протоколи, и основно на IP протокола и тези около него (UDP,TCP,ICMP) е вградена в ядрото, което води до добра производителност в мрежови условия, както и до лесно и удобно ползване на Internet под Linux.

3.2.Адреси и имена, свързване

За да си обменят информация, компютрите трябва да могат да се разпознават един с друг. Основното понятие в това разпознаване е IP-адреса. Всеки компютър в Internet има уникален адрес, за да може да изпраща и да получава информация. По-точно адресира се не самият компютър, а устройството, което го свързва в Мрежата.

Информацията в Мрежата се придвижва на пакети - сравнително малки порции от един компютър към друг, като самият пакет съдържа IP-адреса на изпращача и получателя, но не и на междинните компютри, които пакетът ще посети по пътя.

Протоколът IP (InterNet Protocol) задава правилата за адресиране на компютрите и правилата за отдалечен обмен на пакети между тях.

IP, или Internet Protocol, е протокол от транспортния слой и е гръбнакът на Интернет - всъщност той е протокола с най-голям обхват, защото използва най-различни протоколи от по-горно и по-долно ниво и изгражда най-голямата мрежа, или по-точно, най-голямата съвкупност от свързани мрежи по света.

При IP протокола всяка машина има собствен IP-адрес , представляващ 4-байтово число, което за по-лесна четимост от потребители се записва като 4 еднобайтови числа (т.е. от 0 до 255), разделени с '.' (точка). Пример за такъв адрес е 12.124.12.41. Използвайки адреса на получателя устройства, наречени рутери (междинните компютри по пътя на пакета) вземат решение през коя от другите си връзки да пренасочат пакета.

Всъщност, машините поддържащи IP се делят на 2 вида - рутери и хостове. Рутерите имат 2 или повече интерфейса (т.е. хардуер свързан с различни среди) и изпращат пакети от някой интерфейс към друг, ако не са предназначени за тях самите. Хостовете са машини, които имат само един интерфейс и го използват само за лични нужди. Пример за хост е всеки обикновен компютър свързан към Интернет. Той от своя страна е свързан с рутера на вашия Интернет доставчик, който има повече от 1 мрежови интерфейси - по един за всеки клиент и още 1 (може и повече) за връзка с останалата част от Мрежата.

Предполага се, че Мрежата се състои от свързани помежду си по-малки мрежи (подмрежи). Поради това IP-адресите (и подмрежите) се разделят условно на няколко класа според големината на подмрежата, в която е адреса:

- Клас А е за големи подмрежи, първият байт на IP-адресите в класа е от 1 до 126. Мрежата 127.0.0.0 е специална - 127.0.0.1 значи локалната машина, а 127.0.0.0 - локалната мрежа (за самата машина, не LAN-а; този термин има по-абстрактно значение). За IP-адресите от този клас първият байт е номер на подмрежата, а останалите - номер на на хоста, което позволява да съществуват 126 такива огромни подмрежи с по 255*255*255 хоста всяка.
- Клас Б са адресите с първи байт от 128 до 191 и е за средноголеми подмрежи, за които първите 2 байта са идентификатор на мрежата, а останалите идентификатор на хост-а - това позволява 64*255 подмрежи с 255*255 адреса.
- Клас С мрежите са от 192 до 224, и това са малки мрежи с 3 байта мрежов идентификатор, а останалите - за адрес. Това позволява 32*255*255 мрежи с по 255 адреса.

Останалите адреси с първи байт до 240 (клас D)са за multicasting, а другите са резервирани за бъдеща употреба. Клас А мрежите принадлежат на големи учреждения, които могат да си позволят да използват повече адреси отколкото им позволява Клас Б, и това са 'гиганти' като МИТ например. Клас Б се използва от големи учреждения, които нямат нужда от А мрежа, но и клас С не ги задоволява. Клас С се ползват доста от средните и малки доставчици - зони от 255 адреса, достатъчни за известен брой диал-ьпи (10-20), няколо сървъра, няколко наети линии с подмрежи по 32 адреса и т.н. За хората, които нямат нужда от 255 адреса, може зони от някой клас да се делят на по-малки от например 32 или 64 IP адреса чрез механизма на маските (netmask).

маската прилича на IP адреса по това, че е едно 32 битово число, което обаче определя докъде се простира нечия зона - например адрес 194.52.11.192 с нетмаск-а 255.255.255.224 значи ип адресите от 194.52.11.192 до 194.52.11.224 (т.е. 32 IP адреса). Всъщност, когато конвертираме нетмаската в двоичен вид, тя изглежда по следния начин: 11111111.11111111.11111111.11100000, или последните 5 бита са 0, т.е. 2^5 адреса (2 на степен 5). По същия начин могат да се режат и на зони с по 16, с по 128 или колкото са нужни IP. Клас D се ползва от специални приложения, като тези за видеоконференции, заедно със протокол, наречен IGMP. Също така всеки IP адрес, представялващ адрес на мрежа с последните битове само 0 или само 1 е т.нар. broadcast адрес, който се използва за пращане на информация до всички компютри на дадената мрежа. А специалният адрес 255.255.255.255 представлява т.нар. limited broadcast, който означава 'всички компютри на локалната мрежа' и се ползва обикновено от протоколите BOOTP и DHCP за откриване на сървъри, от които компютър без предварително определен адрес да получи уникален IP-адрес докато се стратира.

Дотук ставаше дума за уникални адреси - те са аналогични на междуградски телефонни номера за набиране, състоящи се от код за набиране на града (номер на подмрежата) и номер на абоната в съответния град. Освен тези уникални номера има уговорка за ползване на част от номерата за вътрешни нужди (по същия начин, както съществуват вътрешни телефонни централи в по-големите фирми). Този тип IP-адреси са предназначени за изграждане на изолирани по някакъв начин локални мрежи (интранет мрежи). Те не са пълноценни IP-адреси и за участие в обмен на данни с друг компютър от Мрежата трябва да бъдат представяни от компютър-посредник, притежаващ истински IP-адрес (този посредник се нарича proxy, firewall или маскиращ сървер в зависимост от реализираната техника за посредничество).

Номерата за интранет мрежи са:

- 10.xx.xx.xx за голяма интранет мрежа

- 172.12.xx.xx до 172.28.xx.xx за средна интранет мрежа
- 192.168.0.xx до 192.168.255.xx за малка интранет мрежа

Пакетът, който пътува в Мрежата по правилата на протокола IP се състои от служебна част и данни. Данните са за протоколите стоящи над IP. Служебната част съдържа следните важни парчета информация:

- адреси на получателя и изпращача
- дължина на пакета
- тип на протокола стоящ над IP, за който е предназначен пакета
- други неща свързани с контрола по транспортирането.

Ако адреса на получателя е собствения ни компютър или компютър пряко свързан с него, Linux ще го изпрати веднага към целта. По-сложно е положението когато получателя е отдалечен компютър. В този случай се задействат алгоритмите за пренасочване (рутиране). Всеки компютър свързан в Мрежата има вътрешна таблица за рутиране. По нея се определя накъде да тръгнат пакетите, за които целта не е пряко достижима. Когато комютърът е свързан в Мрежата, в тази таблица има описан поне един рутер (gateway). Съдържанието и можем да разглеждаме с командата:

```
netstat -r
```

Таблицата за рутиране се променя динамично. Най-прост пример е домашен компютър, който се закача през модем към доставчик на интернет услуги (ISP).

Преди изграждане на връзката таблицата е проста - съдържа информация само за домашния компютър и вътрешния му IP-адрес 127.0.0.1. Всички пакети които се местят ползват за получател и изпращач само този адрес.

След изграждане на връзка към доставчика, компютъра получава още един (вече уникален) IP-адрес за серийния порт, свързан с модема. По новоизградената линия има връзка към IP-адреса на доставчика. Неговия адрес става и рутер за нашия компютър. Като единствен рутер той ще поеме всички изходящи пакети, които не са адресирани към нашия компютър, а също така ще доставя всички входящи пакети.

3.3.Клиенти и сървери

Процесите, които изграждат мрежова връзка помежду си са равностойни по принцип, но на практика се оказва удобно разделянето им на два типа - клиенти и сървери. Приема се, че клиентът е програма, предназначена да ползва мрежови услуги, сървърът - да ги осигурява.

Типичната структура на мрежова работа изглежда така:

На постоянно работещ компютър със статичен IP адрес се стартира програма сървер. Сървърът очаква TCP връзки на предварително определен порт (честота, ако правим аналогия с радиоразпръскването).

При стартиране на програмата-клиент се изпълнява следната последователност:

- адресация - определя се IP адреса на сървера и евентуално порта за връзка.
- свързване - клиентът инициира TCP връзка към сървера.
- обмен - обменят се данните, съобразно спецификата на работата.
- край - TCP връзката се затваря.

Съвместната работа на няколко клиенти с един сървер предполага съществуването на няколко активни TCP връзки. Сървърът трябва да обслужва всички запитвания едновременно, иначе ще се наложи някои от клиентите да чакат и да се ядосват.

Реализацията на този необходим за сървериите паралелизъм се извършва по различни начини. В зависимост от начина им на работа можем да разделим сървериите на няколко типа - fork-сървери , select-сървери и thread-сървери.

fork-сървърът при всяка нова заявка за изграждане на TCP връзка стартира свое копие (извиква системната команда fork), като новото копие поема обслужването на новопоявилия се клиент. Така във всеки момент в системата работят толкова копия на

сървера, колкото са клиентите, очакващи информация. Разбира се, трябва да има и едно излишно копие на сървера (или друга подобна програма), което да следи за появата на нови клиенти.

select-сървърът не стартира паралелни процеси, а сам организира псевдопаралелно обслужване на всичките си клиенти. За да разбере кои от свързаните клиенти му изпращат заявки той ползва системната команда select, откъдето идва името му. Реализацията на select-сървер е трудоемка и сложна, поради необходимостта от реализация на вътрешен паралелизъм, но той е единственото решение, когато обслужването на клиентите изисква синхронизация на извършваните операции. Всички сървери на системи за управление на бази данни са от този тип, за да могат да синхронизират транзакциите в базата данни. Друго предимство е по-високата скорост при голям брой кратки сеанси с клиентите, поради спестяването на времето за изпълнение на бавната команда fork. Например WEB-сървера Apache, който е модифициран fork-сървер е около 3 пъти по бавен при сервиране на статични html-страници от select-сървърите, вършещи същата работа (thttpd, mathpod). Добре написан селект-сървер извършва всички входно-изходни операции в режим nonblock, т.е. ако няма готовност за четене или писане от файл или мрежова връзка, сървера не спира, а върши друга работа.

thread-сървърът представлява процес с няколко активни thread-а ("нишки"), т.е. подпрограми, които работят паралелно с главната програма (или процедури, за запознатите с функционалното програмиране). Когато се получи TCP заявка се стартира нова нишка, която за разлика от fork не представлява копие на програмата и не заема толкова ресурси, тази нишка започва да обработва заявката, докато в същото време главната програма продължава да очаква нови заявки.

Специално внимание заслужава метасървера inetd. Той се стартира при пускането на системата и се ослушва за нови клиенти на различни портове. Когато на някой от прослушваните портове се появи клиент, inetd решава кой сървер съответства на този порт, стартира го и му пренасочва вече изградената мрежова връзка, за да бъде обслужен клиента. След това inetd продължава да чака появата на нови клиенти. Така inetd позволява само с един пуснат сървер да се призовават при нужда няколко други, за да се пестят системните ресурси. Повечето fork-сървери са пригодени за съвместна работа с inetd. Конфигурационният файл /etc/inetd.conf описва кои от тях са подвластни на inetd и специфичните им параметри (порт, тип на връзката с клиента и пр.).