СОФИЙСКИ УНИВЕРСИТЕТ "Св. КЛИМЕНТ ОХРИДСКИ"

ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

КАТЕДРА Изчислителни системи

ЗАПИСКИ

ПО

КОМПЮТЪРНИ МРЕЖИ

Доц. д-р Стела Русева

2019

Настоящото учебно пособие е предназначено за тези студенти, които сега започват за изучават дисциплината "Компютърни мрежи". То въвежда в основните понятия на съвременните компютърни мрежи и комуникации – модели, среди, компоненти, услуги, протоколи, интерфейси, принципи на работа. Целта му е студентите да получат базови знания за компютърните мрежи: за организацията и функционирането им, използваните методи и алгоритми, за основните протоколи и за развитието на системите.

С настоящото учебно пособие се цели да се представят основните концепции по разглежданите въпроси, като не се акцентира на някои детайли, залагайки на обстоятелството, че заинтересованите студенти самостоятелно ще успеят да разширят знанията си, използвайки по-солидни източници или Интернет.

Според мнението на мрежовите специалисти 50% от знанията в тази динамична област на технологиите напълно остаряват за всеки 5 години. С други думи, факт е, че базовите технологии, представите за перспективност или безперспективност на една или друга технология, подходите и методите за решаване на основните задачи, даже и на такива, като създаването на компютърните мрежи – всичко това се променя твърде бързо и често пъти неочаквано.

1. Увод

Изграждането на Интернет е еволюция на идеи.

Компонентите на компютърната мрежа са се създавали в различно време от различни хора за различни цели и са преминали през дълъг и сложен процес на разработване и развитие.

Развитието на компютърните мрежи е започнало в края на 60-те години на миналото столетие и е изключително интересно и наситено.

Първоначално са се използвали големи изчислителни машини с терминали. После за свързването им с други машини (компютри) са започнали да се използват телефонните линии. Но за компютърната мрежата се е изисквал по-различен метод за предаването на данните, от този за предаването на данни по телефона.

Знаете ли какъв е бил телефонът в края на 60-те години на миналото столетие?

По това време три групи инженери, намиращи се на различни части от света, независимо един от друг са започнали да разработват технологии за комутация на пакети, в качеството на мощна и ефективна алтернатива на комутацията на канали (на телефонните линии). Първата научна работа по тази тема е публикувана от Леонард Клейнрок (Leonard Kleinrock). С помощта на теорията на опашките (Queueing Systems) нагледно се демонстрира ефективността на комутацията на пакетите в условия на неравномерно натоварване на линиите.

През 1964 г. Пол Баран (Paul Alexander Baran) започва експерименти с комутацията на пакети в защитени военни мрежи, а Доналд Дейвис (Donald Watts Davies) и Роджър Скентълбъри (Roger Scantlebury) усвояват нова технология в националната физическа лаборатория на Англия.

Тези разработки полагат основата на съвременния Интернет.

Министерството на отбраната на САЩ ръководят проектите за създаването на обединена компютърна мрежа, по-конкретно *Агенцията за съвременни изследователски проекти - Advanced Research Projects Agency (ARPA)*, впоследствие преименувана в DARPA.

Първите протоколи на ARPANET са работили бавно и често са довеждали до срив на мрежовите комуникации. В статията на Винтон Г. Сърф (Vinton G. Cerf) и Робърт Е. Кан (Robert Elliot Kahn) - *A Protocol for Packet Network Intercommunication (списание* IEEE Transactions of Communications, май 1974 г.) е предложен нов набор от основни протоколи.

През 1982 г. TCP/IP става официален протоколен стек в ARPANET.

През 1984 г. броят на свързаните машини в ARPANET превишава хиляда.

През 1986 г. Националният научен фонд на САЩ (National Science Foundation) създава мрежата NSFNET, осигуряваща достъп през Интернет до суперкомпютърните

центрове. След това започва бързо да се разрастват мрежите, свързани с ARPANET. През втората половина на 1980-те години този конгломерат се разглежда като мрежа на мрежите, а после като Интернет. В значителна степен растежът на Интернет е възможен благодарение на свързването на такива мрежи, като SPAN — мрежа на физиката на космоса на NASA, HEP — мрежа на физиката на високите енергии, BITNET — мрежа на машините от клас mainframe на фирмата IBM, EARN — европейска мрежа на научно-изследователските организации.

През 1989 г. се ражда идеята за Световна паяжина (World Wide Web — WWW), инициатор е британският учен Тим Бърнънс-Лий (Sir Timothy John Berners-Lee), който развива идеите за хипертекст, предложени още през 40-те и 60-те години от Буш (Vannevar Bush) и Нелсън (Theodor Holm Nelson).

Втората половина на 1990-те се характеризира с небивал прогрес в областта на Интернет технологиите. Тогава множество компании разработват собствени продукти, свързани с глобалната мрежа.

И днес мрежовите технологии продължават своето стремглаво развитие.

И днес Интернет има богат потенциал за научни изследвания.

Създаването на Интернет може по значимост да се сравни с откриването на огъня или на колелото в човешката история.

Използването на Интернет придобива масов характер. Променя се кардинално и човешкото общуване. Все повече време сме във виртуалното пространство.

Постепенно в нашия живот Интернет става все по-значим. Благодарение на него се появяват практически безгранични възможности във всички сфери на човешката дейност. Така се оказахме нов тип цивилизация.

Интернет ще продължи своето развитие. Естествено главният въпрос е в това, дали той ще способства за претворяване на положителните за хората аспекти, гарантиращи благоденствието на човечеството.

Основни контролиращи органи

Координирането на разработките и поддръжката на Интернет се осъществява от следните организационни структури:

- Internet Activities Board (IAB) – централен орган, включващ два подкомитета: изследователски – IRTF (Internet Research Task Force) и стандартизиращ – IETF (Internet Engineering Task Force), изпълняващ функциите анализ, разработване и приемане на стандартите на Internet мрежата, получили наименованието RFC (Request For Comments);
- Network Information Center (NIC) – орган, отговорен за разпространението на техническата информация, на работата по регистрацията и свързването на

потребителите към Internet и за решаването на редица административни задачи, като разпределението на адресите в мрежата и др.

Internet Corporation for Assigned Names and Numbers (ICANN) контролира системата за имена на домейни DNS - кореновата зона от 2016 г., като преди това са се контролирали от Министерството на търговията в САЩ (the United States Department of Commerce).

Основни положения

Протоколът е глобална договореност на всички разработчици по света за това, как ще се предава информацията.

Съществуват много протоколи, които решават различни задачи. Едни протоколи работят в локални мрежи, други - в глобални мрежи.

Структурата на Интернет (мрежата) може да се представи като множество от компютри, наричани хостове (хост-машини), свързани в единна обединена мрежа, представляваща съвкупност от физически мрежи, наричани подмрежи, съединени с маршрутизатори (рутер, router).

Хост-машина се нарича всеки компютър, свързан към Интернет.

В качеството на подмрежи могат да се използват локални мрежи, работещи под управлението на някои апаратно зависими протоколи (Ethernet, Token Ring), или комуникационни системи с произволна физическа природа (модемни комутируеми линии, X.25 мрежи, Frame Relay, FDDI, ATM и др.). При това всички функции на IP (Internet Protocol, мрежов протокол) се изпълняват от хостовете и маршрутизаторите, наричани възли в мрежата.

Основната цел на мрежата е да се осигури на потребителите на мрежата потенциал за съвместно използване на ресурсите на всички компютри.

Комутация се нарича процесът на доставка на съобщение от един абонат до друг. Организацията на взаимодействието между абонатите на компютърната мрежа се нарича комутация. Комутацията в мрежата може да бъде реализирана с различни механизми, които могат да се класифицират в две групи:

- методи на комутация без междинно съхраняване на данните;
- методи на комутация със съхраняване на данните в междинните възли.

В качеството на метод за комутация без междинно съхраняване на данни в компютърните мрежи се използва комутация на канали, традиционна за телефонните мрежи.

За предаването на данни в компютърните мрежи е бил разработен нов метод за комутация – комутация на съобщения, предполагащ използване в качеството на свързващи възли специализирани средства от изчислителната техника. Това е

позволявало да се реализира в междинните възли съхраняването на предаваните данни и е осигурявало редица предимства, в сравнение с комутацията на каналите. По-нататъшното развитие на методите за комутация е било насочено към усъвършенстването на комутацията на съобщенията, с цел осигуряване на определено качество на предаваните данни.

За свързването на абонатите в компютърните мрежи се използват три метода за комутация: *комутация на канали, комутация на пакети и комутация на съобщения.*

Комутацията на канали, а също така и комутацията на пакети, може да бъде динамична или постоянна.

В мрежите с комутация на канали абонатите се свързат чрез съставен канал, образуван от комутаторите в мрежата при заявка на единия от абонатите.

За съвместно разделяне на каналите между комутаторите на мрежата от няколко абонатни канали се използват две технологии: честотно разделяне на канала (FDM) и разделяне на канала във времето (TDM). Честотното разделяне е характерно при аналоговата модулация на сигналите, а времевото - при цифровото кодиране.

Мрежите с комутация на канали добре комутират потоци от данни с постоянна интензивност, като например потоци от данни, създавани от разговарящи по телефона събеседници. Но те не могат да преразпределят динамично пропускателната способност на магистралните канали между потоците на абонатните канали.

Мрежите с комутация на пакети предават пулсиращ компютърен трафик. Буферизацията на пакетите на различните абонати в комутаторите позволява да се изгладят неравномерностите на интензивността на трафика на всеки абонат и равномерно да се натоварят каналите за свързване между комутаторите.

Мрежите с комутация на пакети работят ефективно, в смисъл, че обемът на предаваните данни от всички абонати в мрежата за единица време е по-голям от този, с използване на мрежи с комутация на канали. Обаче, за всяка двойка абонати пропускателната способност на мрежата може да се окаже по-ниска, от тази с комутация на канали, благодарение на използването на опашки от пакети в комутаторите.

Мрежите с комутация на пакети могат да работят в един от двата режима: дейтаграмен или режим на виртуалните канали.

Размерът на пакета съществено влияе на производителността на мрежата. Често пъти пакетите в мрежата имат максимален размер 1-4 Кбайта.

Комутацията на съобщения е предназначена за организация на взаимодействието на потребителите в off-line режим, когато не се очаква незабавна реакция на съобщението. При този метод на комутация се предполага предаване през няколко транзитни възли, където те изцяло се буферират на диска.

2. Топология на компютърните мрежи

Важна характеристика на мрежата е нейната топология. Тя е от тип граф, върховете на който съответстват на компютрите в мрежата (понякога и друго оборудване, например концентратори), а ребрата са физическите връзки между тях. Конфигурацията на физическите връзки се определя от електрическите съединения между компютрите и може да се отличава от конфигурацията на логическите връзки между възлите в мрежата. Основни топологии при физическите връзки са: напълно свързана, частично свързана, дървовидна, обща шина, кръгова и звезда.

Топологията «обща шина» (bus), представлява кабел, наричан шина или магистрала, към който се свързват хостовете в мрежата. Пакетите, предавани от кой да е хост, заемат шината през цялото време на предаването, при което останалите хостове, имащи пакети за предаване, трябва да изчакат освобождаването на общата шина. Така във всеки момент от време може да предава данни само един хост в мрежата и пропускателната способност на общата шина се разпределя между всички компютри. Основното достоинство на топологията «обща шина» е простата структурна и функционална организация и като следствие, тя е евтина, което я прави най-привлекателна за локални мрежи. Недостатък на тази топология е ниската надеждност на мрежата, като излизането от строя на общата шина води до пълно спиране на мрежата.

Топологията «дърво» (tree) се формира на принципа «минимална сумарна дължина на връзките между възлите в мрежата» и представлява база за построяване на йерархични мрежи. В такива мрежи за предаването на пакетите съществува единствен път между всяка двойка възли, което прави процедурата за маршрутизация тривиална.

Топологията «звезда» (star) съдържа един централен възел, към който се свързват всички останали възли в мрежата. В качеството на централен възел може да работи мощен компютър, към който са присъединени по-малко мощни периферни хостове. В този случай централният компютър може да предоставя своите ресурси (файлове, дисково пространство, ресурси на процесора) на периферните хостове, или да изпълнява функции на маршрутизатор при обмяната на пакети между компютрите в мрежата. Възможна е и друга организация на топологията «звезда», когато в качеството на централен възел се използва мрежово устройство (например концентратор или комутатор), с помощта на който всички хостове са свързани в единна мрежа и което осигурява единствено обмен на данни между хостовете. Ако в качеството на централен възел в мрежата се използва концентратор, тогава логическата топология на мрежата може да бъде както «звезда», така и «обща шина».

В топологията «пръстен» (ring) всеки възел е свързан с други два възела. Пакетите, изпратени от някой възел, преминават през всички други възли на мрежата, като могат и да се върнат при изпращача си. Основното достойнство на тази топология, в сравнение с разгледаните вече топологии, представлява възможността за предаване на пакетите в двете посоки, т. е. възможност всеки възел да има и алтернативен път, по който да могат да бъдат предадени данните при отказ на главния път. При това цената на мрежата за неголям брой на възлите е съизмерима с цената на мрежите с топологии «звезда» и «дърво». Обаче, с увеличаването на броя на възлите в мрежата цената може да се окаже значителна.

«Напълно-свързаната» (full mesh) топология се формира на принципа «всеки с всеки», т. е. всеки възел в мрежата има връзка с всеки друг възел. Такава топология е най-ефективна по всички основни показатели за качество на функциониране: надеждност, производителност и т.н., но поради високата си цена на практика не се използва.

«Много-свързаната» или «клетъчната» топология представлява топология от произволен вид, която се формира на принципа «всеки възел в мрежата е свързан с поне други два възела», т. е. за всеки възел в мрежата винаги има поне един алтернативен път. Такава топология може да бъде получена чрез изтриване в напълно-свързаната топология на някои свързващи линии (например тези, които не се използват за предаване на пакетите или са малко натоварени), което често пъти съществено намалява цената на мрежата.

«Смесената» топология представлява всякаква комбинация от разгледаните вече топологии и се образува обикновено при обединяването на няколко локални мрежи, например три мрежи с топология «звезда» се свързват в мрежа с топология «пръстен».

Логическите връзки представляват маршрутите на предаването на данните между възлите на мрежата. Като средства за логическа структуризация се използват мостове, комутатори, маршрутизатори и шлюзове.

Да се избере маршрута за предаването на пакетите означава да се определи последователността на транзитните възли и техните интерфейси, през които трябва да се предадат пакетите, за да достигнат до получателя.

Според размера си или териториалния си обхват се различават локални мрежи (LAN), глобални мрежи (WAN) и градски или регионални (MAN) мрежи.

Според режима на предаване се различават:

- Предаване до всички (общодостъпно – Broadcast). Прилага се в LAN. Използва се общ комуникационен канал, който се разпределя между всички в мрежата. Пакетите (съобщенията) се получават от всички, но ги прочита само този, който си познае адреса. Частен случай е груповото предаване (multicast).
- Предаване точка-точка (Point-to-point) – WAN мрежите се състоят от множество връзки “точка - точка” с произволна топология. Затова се налага маршрутизация – намиране на оптималния път.

Свързващата линия представлява физическата среда за предаване, по която се предават сигналите, заедно с апаратурата за предаване на данни, формираща сигналите, съответстващи на типа на свързващата линия.

3. Характеристики на мрежите

Качеството на работата на мрежата се определя от следните характеристики: производителност, надеждност, съвместимост, управляемост, защитеност, разширяемост и мащабируемост.

Разширяемост (extensibility) - възможност за сравнително лесно добавяне на отделни елементи на мрежата (потребители, компютри, приложения, услуги), увеличаване на дължината на сегментите на мрежата и замяна на съществуващата апаратура с по-мощна.

Мащабируемост (scalability) - мрежата позволява да се увеличава броя на възлите с дължина на свързванията в много широки граници, при което производителността на мрежата не намалява.

Прозрачност (transparency) – свойство на мрежата да се скриват от потребителя детайлите на вътрешното устройство, като така се опростява неговата работа в мрежата.

Управляемост на мрежата (Controllability) - подразбира се възможност за централизиран контрол на състоянието на основните елементи на мрежата; откриване и разрешаване на проблемите, възникващи при работа на мрежата; анализ на производителността и планиране на развитие на мрежата.

Съвместимост (интегрируемост) - мрежата е способна да включва в себе си най-разнообразно програмно и апаратно осигуряване.

За оценяването на надеждността на мрежите се използват най-различни характеристики, в това число: коефициент на готовност (availability), определящ времето, в течение на което системата може да бъде използвана; безопасност (security), т. е. способност на системата да защити данните от несанкциониран достъп; отказоустойчивост (fault tolerance) – способност на системата да работи в условия на отказ на някои от нейните елементи.

Под архитектура на компютърната мрежа се разбира множеството от технически и инженерни решения за структурната и функционалната организация на мрежата, осигуряващи определена съвкупност от нейни свойства и характеристики, разглеждани от гледна точка на потребителя на мрежата и отличаващи дадената конкретна мрежа от всяка друга мрежа.

Под технология на компютърната мрежа (мрежова технология) се разбира съвкупността от методи на организация (реализация) на предаването и обработването на данните, осигуряващи достигането на определени цели, формулирани във вид на изисквания към качеството (ефективността) на обработването и предаването на данните.

Пропускателната способност (throughput) на линията определя максималната възможна скорост за предаване на данните по свързващите линии. Пропускателната способност на линията се измерва в битове в секунда - (bps), а също така и в производни единици, като килобит в секунда (Kbps), мегабит в секунда (Mbps), гигабит в секунда (Gbps) и т. н.

Пропускателната способност на свързващата линия и комуникационното мрежово оборудване традиционно се измерва в битове за секунда, а не в байтове за секунда. Данните в мрежите се предават последователно, т. е. побитово, а не паралелно (побайтово), както това е реализирано в компютъра между компонентите. Такива единици за измерване, като килобит, мегабит или гигабит, в мрежовите технологии строго съответстват на степените на 10, т. е. килобит – това са 1000 бита, а мегабит – това

са 1 000 000 бита, както е прието във всички сфери на науката и техниката, а не близки до тези числа, които са степени на 2, както е прието в програмирането, където приставката кило е равна на 2 на 10 =1024, а мега - 2 на 20 = 1 048 576.

4. Еталонен модел на мрежите и TCP/IP

Организацията на взаимодействието между устройствата в мрежата е сложна задача. Както е известно, за решаването на сложни задачи се използва универсален метод – *декомпозиция*, т.е. разбиване на сложна задача на няколко по-прости задачи – модули.

Декомпозицията се изразява в точното определяне на функциите на всеки модул, а също така и тяхното взаимодействие (интерфейси).

Като резултат се постига логическо опростяване на задачата и се появява възможност за автономно разработване и модифициране на отделните модули без да се променя останалата част на системата.

При декомпозицията често пъти се използва многослойния подход. Цялото множество от модули, решаващи частни задачи се разбива на групи и се подрежда в нива (слоеве), образуващи йерархия. В съответствие с принципа на йерархията, за всеки междинен слой могат да се посочат непосредствените му съседни - горе и долу лежащи.

Такава йерархична декомпозиция на задачите предполага ясно определяне на функциите на всеки слой и на интерфейсите между нивата. Интерфейсът определя набора от функции, които долу лежащия слой предоставя на горе лежащия. В резултат на йерархичната декомпозиция се постига относителна независимост на нивата и възможност за автономно разработване и модифициране.

OSI моделът (Open Systems Interconnection model) стандартизира броя, функциите и предназначението на нивата на системните средства за взаимодействие. OSI стекът конкретизира зададен набор от протоколи.

Формализираните правила, определящи последователността и формата на съобщенията, които си обменят мрежовите компоненти, лежащи на един слой, но на различни възли, се наричат протокол.

Слой (layer) е понятие, позволяващо да се раздели цялата съвкупност от функции, свързани с обработването и предаването на данните в компютърните мрежи на няколко йерархически групи. На всеки слой се реализират определени функции по обработването и предаването на данните с помощта на апаратни и/или програмни средства на мрежата. Всеки слой обслужва по-горе лежащия слой и, на свой ред, използва услугите на по-долу лежащия.

При пакетната комутация се добавят към данните служебни заглавия, например с информация за това, от кого са данните, за кого са предназначени и така се изпращат в мрежата. По отношение на размера на изпращаните данни се търси някаква златна среда,

така че те да не са прекалено малко или твърде много и да не нарушават стабилната работа на системата. Решението за това, колко данни да се разположат в пакета, как да се опаковат тези данни или каква служебна информация да се добави – получава наименованието протокол.

МОДЕЛ OSI/RM

Стек означава набор от свързани протоколи, процедури, стандартизирани правила, обединени логически вертикално. Данните се предават отгоре на долу при изпращане в средата за предаване, или отдолу нагоре при получаване от програмното осигуряване на горния слой.

Физически (PHYSICAL)

Определя механическите, електрическите, функционалните и процедурните характеристики на физическото съединение.

Канален (DATA LINK)

Задава средствата за установяване, поддръжка и освобождаване на линиите за предаване на пакетите с данни.

Мрежов (NETWORK)

Определя маршрута на предаването на пакетите.

Транспортен (TRANSPORT)

Осигурява комутационно обслужване, прозрачно предаване на пакетите между абонатите (доставка без грешки, загуби, дублиране).

Сесиен (SESSION)

Управлява диалога между обектите в мрежата (сеанса), синхронизация на съобщенията.

Интервалът от време, в течение на който взаимодействат процесите, се нарича сеанс или сесия.

Презентационен (PRESENTATION)

Управлява информационния обмен, кодирането и декодирането на данните.

Приложен (APPLICATION)

Поддържа приложните процеси на крайния потребител, управлява взаимодействието на тези програми с обектите на мрежата.

Табл.1 **Съответствие на OSI и на другите протоколни стекове**

Ниво на OSI		Класифика-ция	TCP/IP (DoD) стек	Класификация	Примерни протоколи	Единици за измерване	Устройства
7	Приложен слой	Ориенти-ран за приложе-нието	Приложение	End-to-end (много хопово)	HTTP FTP HTTPS SMTP LDAP NCP	Съобщения	Шлюз, Комутатор на 4-7 слой
6	Представителен						
5	Сеансов						
4	Транспортен	Ориенти-ран за предаване	Транспорти-ране		TCP UDP SCTP SPX	TCP сегменти UDP дейтаграми	
3	Мрежов		Маршрути-зация	Точка-точка и broadcast	ICMP IGMP IP IPsec IPX	Пакети	Маршрутизатор, комутатор на трети слой
2	Контрол за грешки + адресация (Data Link)		Достъп до мрежата		Ethernet Token Ring FDDI ARCNET	Кадри (фреймове, frames)	Мост (bridge), комутатор (switch)
1	Предаване на битове (Физически)					Битове	Повторител (repeater), концентратор (hub)

Докато комитетите на ISO са съгласували своите стандарти, се променя цялата концепция на мрежите и по цял свят се внедрява TCP/IP.

Днес OSI постепенно се движи към състояние да се окаже на страниците на историята на компютрите.

Други разработени стекове

Стек IPX/SPX

Това е оригинален стек от протоколи на фирмата Novell, разработен за собствената мрежова операционна система NetWare в началото на 80-те години. Протоколите Internetwork Packet Exchange (IPX) и Sequenced Packet Exchange (SPX), които дават името на стека, са адаптация на протоколите XNS на фирмата Xerox, разпространени в много по-малка степен от IPX/SPX.

Стек NetBIOS/SMB

Фирмите Microsoft и IBM съвместно разработват стека протоколи NetBIOS/SMB. Средствата NetBIOS се появяват през 1984 г. като мрежово разширение на стандартните функции на базовата система за вход/изход (BIOS) IBM PC за мрежовите програми PC Network на фирмата IBM, която на приложния слой се е използвала за реализация на мрежовите услуги на протокола SMB (Server Message Block).

През 90-те години става ясно, че конкуренцията при мрежите следва да се замени с тяхното обединение. Така възниква Интернет. От обединението печелят както производители, така и ISP (повече информационни ресурси, повече услуги, повече клиенти).

Следва да се различават моделът OSI и стекът OSI. Моделът OSI е концептуална схема за взаимодействие на отворени системи, а стекът OSI е набор от конкретни протоколи. За разлика от протоколните стекове, стекът OSI напълно съответства на модела OSI.

Протоколите на стека OSI са сложни и с неоднозначна спецификация на свойствата. Те са резултат от общата политики на разработчиците на стека, които са се стремили в своите протоколи да отчетат всички възможни варианти и всички съществуващи и появяващи се технологии. Също така са направени голям брой политически компромиси, неизбежни при приемането на международни стандарти по такъв злободневен въпрос, като построяване на отворени изчислителни мрежи.

Отворени, в смисъл - публикувани, общостъпни спецификации, съответстващи на стандартите и приети в резултат на достигнато съгласие след всестранно обсъждане от всички заинтересовани страни.

Стекът OSI е международен, независим от производителите стандарт.

Един от най-големите производители, поддържащ OSI, е компанията AT&T. Нейната мрежа Stargroup се базира на този стек.

TCP/IP е събирателно название за набор (стек) от мрежови протоколи от различни слоеве, използвани в Интернет.

Особености на TCP/IP:

- Отворени стандарти на протоколите, разработвани независимо от програмното и апаратното осигуряване;
- Независимост от физическата среда за предаване;
- Система за уникална адресация;
- Стандартизирани протоколи от най-високия слой за разпространяване на потребителските услуги.

5. Структура на мрежовите пакети

При формирането на пакета, изпращан в мрежата, всеки слой добавя към байтовете, постъпили от по-горе лежащия слой, свои служебни данни във вид на заглавия или трейлъри.

На приемащата страна при движението на данните от долу нагоре по протоколния стек съответстващото програмно осигуряване анализира заглавието на своя слой и предава вложените в него данни на програмните средства на по-горе лежащия слой.

Данните, предавани на различните слоеве в мрежата, се формират във вид на блокове, наричани протоколни блокове с данни (Protocol Data Unit – PDU). PDU представлява единица от данни, предавани като единно цяло и притежаващи допълнителна опаковка във вид на заглавие със служебна информация (адрес на изпращача, адрес на получателя, дължина на блока с данни и т.н.) и евентуално възможна опашка.

На различните слоеве на OSI модела се използват различни PDU, притежаващи специални наименования. Най-широко разпространение са получили следните наименования на блоковете с данни: съобщение, дейтаграма, пакет, кадър (фрейм) и сегмент.

Съобщение (message) – блок от данни, разглеждани като единно цяло при предаването между двама потребители (процеси) и имащи определено смислово съдържание. Съобщенията обикновено се използват на 7 слой в OSI модела за предаването на данни между приложните процеси и могат да имат произволна дължина.

Кадър (frame) – блок от данни на 2-ри (каналния) слой на OSI модела, притежаващ ограничена дължина и предавани като единно цяло в локалната мрежа или по предоставен канал за връзка между два възела.

Поток (stream) се наричат данните, постъпващи от приложенията към протоколите от транспортния слой (TCP и UDP).

В TCP/IP протоколния стек блокът от данни на TCP протокола се нарича сегмент, който се получава чрез изрязване от неструктуриран поток от байтове, постъпващи към TCP протокола в рамките на логическо съединение от протоколите на по-горния слой.

Единицата от данни на UDP протокола често пъти се нарича дейтаграма (datagram).

Дейтаграма е общо наименование за единица от данни, с които оперират протоколите без установяване на съединение. Към тези протоколи принадлежи и IP.

Дейтаграмата на IP протокола също така се нарича и пакет (packet).

Може да се отбележи, че в ATM мрежите данните се предават във вид на блокове с фиксиран размер от 53 байта, които се наричат клетки (cell).

Максималният размер на кадъра, пакета и дейтаграмата зависи от мрежовата технология и се задава от съответните протоколи, определящи формата и допустимия размер на блока от данни.

6. Физическо ниво

1-ят слой – физически (physical layer) е най-ниският слой на OSI модела, определящ процеса на преминаването на сигналите през средата за предаване междумрежовите устройства (възли в мрежата).

Той реализира управлението за свързване на канала:

- включването и изключването на свързващия канал;
- формирането на предаваните сигнали.

Също така описва:

- механическите, електрическите и функционалнте характеристики на средата за предаване;
- средствата за установяване, поддръжка и затваряне на физическото съединение.

При необходимост осигурява:

- кодирането на данните;

- модулацията на сигнала, предаван по физическата среда.

Данните на физическия слой представляват поток от битове (последователност от нули или единици), кодирани във вид на електрически, оптични или радио сигнали.

Поради възникването на шумове, въздействащи на електрическата свързваща линия, достоверността на предаването, измервана като вероятност за изкривяване (Distortion) на един бит, е в диапазона 10^{-4} – 10^{-6} . Това означава, че средно за 10000 – 1000000 бита предавани данни един бит се оказва изкривен.

На физическия слой за разделянето на потока от битове, съответстващи на различни блокове от данни на 2 слой (кадри), могат да се използват различни методи:

1) посочване в заглавието на кадъра на неговата дължина и преброяване на броя на символите в процеса на приемането на потока от данни (основен недостатък – неустойчивост към шумове);
2) използване в качеството на граница за кадрите забранени сигнали на физическия слой;
3) използване в качеството на граници за кадрите специални стартови и стопови символи (байтове);
4) използване в качеството на граници за кадрите специални последователности от битове.

Свързващи линии

При построяването на мрежите се използват свързващи линии, с различна физическа среда: телефонни и телеграфни проводници, медни коаксиални кабели, медни усукани двойки, влакнесто-оптични кабели, радиовълни.

Свързващите линии могат да използват, освен кабели, и междинна апаратура, прозрачна за потребителите. Междинната апаратура изпълнява две основни функции: усилва сигналите и осигурява постоянна комутация между двойка потребителски линии.

В зависимост от типа на междинната апаратура свързващите линии се делят на аналогови и цифрови. В аналоговите свързващи линии за уплътняване на нискоскоростни канали на абонатите в общ висикоскоростен канал се използва метода за разделяне на честотите (FDM), а при цифровите – метода за разделяне във времето (TDM).

С цел определяне на способността на линията да предава сигнали с произволна форма без значителни изкривявания, се прилагат редица показатели, използващи в качеството на тестов сигнал синусоиди с различна честота. Към тези показатели се отнасят: амплитудно-честотна характеристика, честотна лента и затихването на сигнала на определена честота.

В компютърните мрежи се използват кабели, удовлетворяващи определени стандарти. Съвременните стандарти определят характеристиките не на отделен кабел, а на пълен набор от елементи, необходим за създаването на кабелно съединение - например шнур от работната станция до розетката, самата розетка, основния кабел,

твърдо cross-over съединение и шнура до концентратора. Днес най-често използваните стандарти са: американският стандарт EIA/TIA-568A, международният стандарт ISO/IEC 11801, европейският стандарт EN50173, а също така и фирменият стандарт на компанията IBM.

Стандартите са определени за четири типа кабели: на основата на неекранирана усукана двойка, на основата на екранирана усукана двойка, коаксиален и влакнесто-оптичен кабел.

Кабелите на основа на неекранирана усукана двойка в зависимост от електрическите и механическите характеристики се разделят на 5 категории. Кабелите от категория 1 се прилагат там, където изискванията към скоростта на предаване са минимални. Главната особеност на кабелите от категория 2 е способността да предават сигнали със спектър до 1 MHz. Кабелите от категория 3 са широко разпространени и са предназначени както за предаване на данни, така и за предаване на глас. Кабелите от категория 4 представляват подобрен вариант на кабелите от категория 3 и на практика се използват рядко. Кабелите от категория 5 са били специално разработени за поддръжка на високоскоростните протоколи FDDI, Fast Ethernet, 100VG-AnyLAN ATM и Gigabit Ethernet.

Кабелите на основата на екранирана усукана двойка добре защитават предаваните сигнали от външни шумове, а потребителите на мрежите - от вредните за здравето излъчвания. Наличието на заземяващо екраниране оскъпява кабела и усложнява неговото полагане. Екранираният кабел се използва само за предаване на данни. Основният стандарт, определящ параметрите на екранираната усукана двойка, е фирменният стандарт на IBM. В този стандарт кабелите се делят на типове: Type 1, Type 2, ..., Type 9, като основен е кабел от Type 1.

Коаксиалните кабели съществуват в голям брой варианти: «дебел» коаксиален кабел, различни разновидности на «тънък» коаксиален кабел, който притежава по-лоши механически и електрически характеристики в сравнение с «дебелия» коаксиален кабел, но за сметка на своята гъвкавост е по-удобен за монтаж. Такъв е и телевизионният кабел.

Влакнесто-оптическите кабели притежават отлични електромагнитни и механически характеристики, недостатъкът им е в сложността и високата цена на монтажните работи.

Методи за предаване на дискретни данни на физическия слой

При предаване на дискретни данни по нисколентов кабел за тонална честота, използван в телефонията, най-подходящи се оказват вариантите за аналогова модулация, при които носещата синусоида се модулира от изходната последователност от двоични цифри. Тази операция се реализира от модемите.

За нискоскоростно предаване на данни се прилага изменение на честотата на носещата синусоида. Високоскоростните модеми работят на комбинирани варианти за квадратурна амплитудна модулация (QAM), за която са характерни 4 нива на амплитудата на носещата синусоида и 8 нива на фазата. Не всички от 32 възможи

съчетания на QAM метода се използват за предаване на данни, забранените съчетания позволяват да се разпознаят повредените данни на физическия слой.

На широколентовите канали за връзка се прилагат потенциални и импулсни методи за кодиране, в които данните са представени с постоянен потенциал на сигнала на различни нива или с полярностите на импулса или на неговия фронт.

При използването на потенциалните кодове важно значение получава задачата за синхронизация на приемника и предавателя, понеже при предаването на данни, дълги последователности от нули или единици, сигналът на входа за приемника не се променя. На приемника е сложно да определи момента на фиксиране на поредния бит на данните.

Най-популярният импулсен код е манчестърският код, в който информация носи посоката на движение на сигнала в средата на всеки такт. Манчестърският код се прилага в Ethernet технологиите.

Подобрените потенциални кодове притежават по-тесен спектър от импулсните, затова те се прилагат при високоскоростните технологии, като FDDI, Fast Ethernet, Gigabit Ethernet.

7. Канално ниво

Каналният слой или слоят за предаването на данните (data link layer) е вторият слой в OSI модела.

Той реализира управлението на:

- достъпа на мрежовите устройства до средата за предаване, когато две или повече устройства могат да използват една и съща среда за предаване;
- надеждно предаване на данните по каналите за свързване, позволяващо увеличаването на достоверността на предаването на данните на 2-4 порядъка.

Той описва методите за достъп на мрежовите устройства до средата за предаване, основани например на предаването на маркер или на конкуренция.

Той осигурява:

- функционални и процедурни средства за установяване (отваряне), поддържане и затваряне на съединението;
- управление на потока, с цел предотвратяване на препълването на приемащото устройство, ако неговата скорост е по-ниска, от тази на предаващото устройство;
- надеждно предаване на данните по физическия канал с вероятност за повредени данни от 10^{-8} до 10^{-9} за сметка на прилагане на методи и средства за контрол на предаваните данни и повторно предаване на данните при откриване на грешка.

По този начин, каналният слой осигурява достатъчно надеждно предаване на данните през ненадежден физически канал.

Блокът от данни, предавани на каналния слой, се нарича кадър (frame).

На каналния слой се появява свойството адресуемост на предаваните данни във вид на физически (машинни) адреси, наричани още MAC адреси и представляващи обикновено уникални идентификатори на мрежови устройства.

Универсалните MAC адреси са 6 байтови и се записват в шестнадесетичен вид, като байтовета на адреса се разделят с малко тире (дефис), например: 00-19-C5-A2-B4-DE.

Към процедурите на каналния слой работят следните механизми:

- добавяне в кадрите на съответстващите адреси;
- контрол за грешки;
- при необходимост, повторно предаване на кадрите.

Управление на трафика на каналния слой

На каналния слой управлението на потока в канала на връзката между два възела се реализира чрез прилагане на:

- механизъм за използване на положителни и отрицателни квитанции за потвърждение, че е получен кадъра;
- механизъм за таймаут (тук освен квитанции се използва и таймер);
- механизъм на плъзгащия (плаващ) прозорец.

Първите два механизма се ползват при полудуплексен канал, а последният - при пълен дуплекс.

Методи за предаване на данните на каналния слой

Основната задача на протоколите от каналния слой е доставката на кадъра до възела получател в мрежата с определена технология и достатъчно проста топология.

Асинхронните протоколи са се разработвали за обмен на данни между нискоскоростни старт-стопни устройства: телетайпи, алфавитно-цифрови терминали и т. н. В тези протоколи за управлението на обмена на данните се използват не кадри, а отделни символи от долната част на ASCII или EBCDIC кодовите таблици. Потребителските данни могат да се оформят в кадри, но байтовете в тези кадри винаги са отделени от стартови и стопови сигнали.

Синхронните протоколи изпращат кадри както за потребителските данни, така и за управлението на обмена.

В зависимост от начина на определяне на началото и на края на кадъра синхронните протоколи се делят на символно-ориентирани и битово-ориентирани. В първите за тази цел се използват символи от ASCII или EBCDIC кодове, а в последните - специален набор от битове, наричан флаг. Битово-ориентираните протоколи по-рационално изразходват полето с данни на кадрите. С цел недопускане на съвпадението на данните с флага, добавят към него единствен бит, а символно-ориентираните протоколи добавят цял символ.

В дейтаграмните протоколи отсъства процедура за предварително установяване на съединение и затова спешните данни се изпращат в мрежата без забавяне.

Протоколите с установяване на съединение могат да притежават редица допълнителни свойства, отсъстващи при дейтаграмните протоколи. Най-често в тях е реализирана способността за възстановяване на повредени и изгубени кадри.

Едни от най-популярните методи за откриване на грешки са основани на циклически излишни кодове (CRC), които откриват многократни грешки.

За възстановяването на кадрите се използва метода за повторно предаване на база на квитанции. Този метод работи по алгоритъма с изчакване от източника, а също така и по алгоритъма на плаващия прозорец.

С цел повишаване скоростта на предаване на данните, в мрежата се прилага динамична компресия на данните, използваща различни алгоритми. Коефициентът на компресия зависи от типа на данните и използвания алгоритъм и може да варира от 1:2 до 1:8.

Видове протоколи на каналния слой

1) С асинхронен / със синхронен режим на предаване на байтовете;
2) Символно-ориентиран / битово-ориентиран;
3) С предварително установено съединение / дейтаграмен;
4) С откриване на повредени данни / без откриване;
5) С откриване на изгубени данни / без откриване;
6) С възстановяване на изгубени и повредени данни / без възстановяване;
7) С поддръжка на динамична компресия на данните / без поддръжка.

Институтът на инженерите по електроника и електротехника (Institute of Electrical and Electronics Engineers – IEEE) е предложил вариант за OSI модели, използван при разработването и проектирането на локални мрежи и получил наименованието IEEE модел.

В IEEE модела каналният слой се разбива на два подслоя:

- подслой за управление на достъпа до средата за предаване (Medium Access Control, MAC подслой), описващ метода на достъп на мрежовото устройство до средата за предаване на данни;

- подслой за управление на логическото съединение (Logical Link Control, LLC подслой), описващ метода за отваряне и затваряне на съединението, а също така и метода за предаването на данните.

LLC подслоят предоставя на по-високите слоеве възможност да управляват качеството на услугите и да осигуряват обслужване на следните три типа:

1) обслужване без установяване на съединение и без потвърждаване на доставката;
2) обслужване без установяване на съединение с потвърждаване на доставката;
3) обслужване с установяване на съединение.

Обслужването без установяване на съединение и без потвърждаване на доставката не гарантира доставката на данните. Обикновено се прилага в приложенията, които използват за контрол на предаваните данни и защита от грешки протоколи от по-високите слоеве.

Обслужването с установяване на съединение осигурява надежден обмен на данните.

Главната функция на MAC подслоя е осигуряването на достъп до канала за предаване на данни. На този слой се формира физическия адрес на устройствата, който се нарича MAC адрес. Всяко устройство в мрежата се идентифицира с този уникален адрес, който се присвоява на всички мрежови устройства.

8. Канално ниво в локалните мрежи

Ethernet технология

Локалните мрежи работят на каналния слой.

Ethernet е технология (мрежова архитектура) на локални изчислителни мрежи, описана в стандартите на физическия и каналния слой на OSI/RM модела. При построяването на мрежа с комутатори и репитери (повторители, хъбове) Ethernet се изгражда по физическа топология "звезда". Логическата топология на тази архитектура не зависи от кабелната система и винаги е "шина" (в случай на използване на CSMA/CD, като метод за достъп до средата за предаване).

Скоростта на предаването на данните се определя от спецификацията и може да бъде 10 Mb/sec, 100 Mb/sec (Fast Ethernet), 1 Gb/sec (Gigabit Ethernet), 10 Gb/sec (10 Gigabit Ethernet), 40 Gb/sec (40 Gigabit Ethernet), 100 Gb/sec (100 Gigabit Ethernet), 400 Gb/sec (400 Gigabit Ethernet). Във всяка спецификация съществуват още няколко подвида (например: 100Base-TX, 100Base- FX за Fast Ethernet), характеризиращи се с различен вид на свързване към средата за предаване (оптично влакно, усукана двойка, коаксиален

кабел), а също така и различни методи за кодиране на сигнала и за включване/ изключване на едни или други комуникационни опции.

Както вече беше споменато, на каналния слой всички устройства имат свой адрес, обикновено зададен апаратно. В Ethernet технологията в качеството на адрес се използва 6 байтов MAC идентификатор (medium access control, например 00:00:C0:5E:84:0E).

Различават се предаване до всички (broadcast), уникални (unicast) MAC адреси и MAC адреси за групово предаване (multicast). Първият се състои само от 1-ци - 48 разряда (FF:FF:FF:FF:FF:FF). MAC адресът за групова адресация съдържа 1-ца в най-старшия бит на най-старшия байт, като при уникаст адресация той е винаги 0. Останалите битове могат да приемат всякакви стойности.

В качеството на алгоритъм за достъп до средата за предаване се използва метода CSMA/CD (Carrier Sense Multiple Access with Collision Detection, множествен достъп с откриване на носещата /честота/ и разпознаване на колизии – МДОН/РК).

Всички станции, намиращи се в границите на един комутационен възел, прослушват мрежата (логическа топология "шина") и са равноправни по отношение на момента за начало на предаване, което может да започне само при отсътствие на сигнал от другите станции. Моментът за започване на предаването с нищо не е регламентиран, такъв метод за достъп спада към категорията на недетерминираните.

CSMA/CD алгоритъм

1. Преди да започне да предава станцията трябва да се определи дали е "свободна" средата за предаване (като се прослушва носещата честота).
2. Ако не е открит чужд сигнал в средата, станцията може да започне да предава.
3. По време на започването на предаването станцията също така прослушва мрежата, за да определи дали има колизия (изкривена форма на сигналите поради наслагване), която може да се случи вследствие на множествения достъп и почти едновременното започване на предаване от две и повече станции. Ако е открита колизия, станцията изпраща в мрежата специален "jam" сигнал, подпомагащ откриването на колизии от другите станции, а също така прекратява да изпраща данни и преминава в режим на очакване.
4. Периодът на изчакване се определя по случаен начин и неговата продължителност зависи от броя на последователно случилите се колизии.
5. След излизане от режим на изчакване станцията отново може да започне да предава (преход към 1).

Методът на достъп до средата Ethernet мрежите (CSMA/CD) има конкурентен характер. Станциите прослушват средата, и ако тя е свободна, имат права да започнат да предават. Поради неопределеността на този момент и достатъчно голямата дължина на кабелната система, в средата могат да възникнат колизии (т.е. "стълкновения" на сигналите). Това води до необходимост от повторно предаване след някакъв, избран по случаен начин, интервал от време. Колизиите намаляват общата пропускателна способност на мрежата.

Домейн на колизиите е обединена част на кабелната система, станции и друго комуникационно оборудоване, в които е възможно образуване на колизии, или част от Ethernet мрежа, в която няма устройства, буферизиращи кадрите (например: комутатори с проверка за коректност на получения кадър), или множество от всички станции в мрежата, едновременното предаване на всяка двойка, от които води до колизии.

Формат на Ethernet кадъра

Съществуват 4 различни видове Ethernet кадри. Техният общ вид е представен по-долу.

DA(6) SA(6) L/T(2) CRC(4) Data(46-1500)

- DA – адрес на получателя (Destination Address, 6 байта);
- SA – адрес на източника (Source Address, 6 байта);
- L/T – дължина или тип на кадъра (Length/Type, 2 байта);
- Data – данните от по-горе лежащия слой (например за IP слоя е от 46 до 1500 байта);
- CRC – контролна сума (Cyclical Redundancy Check, 4 байта).

Необходимо е да се отбележи, че пред всеки кадър станцията на изпращача добавя преамбула и начален ограничител за кадъра (8 байта). Също така, всички станции трябва да поддържат междукадрови интервали за 96 такта (например 9.6 μsec при битова скорост 10 Mb/sec).

Други протоколи на каналния слой

Протоколите SLIP (Serial Line Internet Protocol) и PPP (Point to Point Protocol) осигуряват свързване към комутируема линия чрез канали за предаване на данните.

SLIP (Serial Line IP) е първият стандарт на каналния слой за заделени линии, разработен специално за TCP/IP протоколния стек, а поради простотата си може да се прилага както при комутируемите, така и за заделените канали. Но SLIP се поддържа само от IP протокола на мрежовия слой.

SLIP протокол позволява в качеството на свързващи линии да се използват последователни телефонни линии. Програмното осигуряване, реализиращо работата с този протокол, първо приема символите, пристигащи от устройството за последователно предаване на данните. После тези данни се разглеждат като съдържимо на IP пакет, след което се опаковат данните в IP пакет и се предават на TCP модула. Реализира се и обратната процедура, SLIP получава от TCP модула IP пакет, отделя неговото съдържимо, форматира го, разделя го на символи и ги изпраща чрез устройство за последователно предаване в мрежата.

HDLC (High-level Data Link Control Procedure) е протокол от високо ниво за управление на канала. Това е стандарт на ISO за заделени линии, представящ LAP (Link

Access Protocol) семейството протоколи. HDLC се отнася към битово ориентираните протоколи.

PPP (Point-to-Point Protocol) е протокол за двуточково съединение, дошъл на смяна на SLIP протокола и построен на база на формата на кадрите на протоколите от семейството на HDLC с допълване със собствени полета. PPP е стандартен протокол на Интернет и също така, както HDLC протокол, представлява семейство от протоколи.

PPP е аналогичен на SLIP съвременен протокол, който може да предава не само IP пакети, но и пакети на IPX; PPP има вградена в протокола автентификация; PPP поддържа динамично задаване на IP адреси; PPP предава по-малко служебна информация, от колкото SLIP, с което се увеличава скоростта на предаване.

Мрежови комутатори

Създаването на сложна структурирана мрежа, интегрираща различни базови технологии, може да се осъществи със средствата на каналния слой: за тази цел се използват някои видове мостове и комутатори. Мостът или комутаторът разделя мрежата на сегменти, като така локализира трафика вътре в сегмента, което прави свързващите линии разделяеми предимно между станциите на дадения сегмент. Така мрежата се разделя на отделни подмрежи, от които могат да бъдат построени съставни с достатъчно големи размери.

Комутаторите (switch) могат да обединяват сегменти с различни технологии на локални мрежи, като транслират протоколите на каналния слой в съответствие с IEEE 802.1H спецификацията. Единственото ограничение за транслацията представлява използването на MTU с еднакъв размер в съединяваните сегменти.

Суичовете поддържат разнообразни потребителски филтри, базирани на MAC адресите, а също така и на съдържимото на полетата на по-горе лежащите протоколи. В последния случай администраторът трябва да изпълни голям обем ръчна работа, свързана с определянето на позицията на полето относно началото на кадъра и неговата проверявана стойност. Обикновено филтрите допускат комбинация от няколко условия с помощта на логическите оператори AND и OR.

Суичовете осигуряват поддръжка на качеството на обслужване с помощта на приоритетна обработка на фреймовете. Стандартът 802.1p определя допълнително поле от 3 бита за съхраняване на приоритета на кадъра, независимо от каналната технология на мрежата.

С цел автоматично поддържане на резервни връзки в сложни мрежи в комутаторите е реализиран алгоритъмът на покриващото дърво - Spanning Tree Algorithm. Този алгоритъм се базира на периодична генерация на служебни фреймове, с помощта на които се откриват и блокират циклични свързвания в мрежата. Протоколът на покриващото дърво (STP), при наличие на излишни физически съединения проектира логическите маршрути така, че логическата топология на мрежата да бъде дървовидна.

Технологията на виртуалните локални мрежи (VLAN) позволява в мрежа, построена с комутатори, да се създадат изолирани групи от възли, между които да не се предава различен тип трафик, в това число и общодостъпен (броудкаст). Виртуалните мрежи имат предимство пред физическите изолирани сегменти в гъвкавостта на реализацията, която може програмно да се променя.

Многослойните комутатори допълнително поддържат маршрутизируем порт, който е интерфейс от 3-ти слой на OSI. Те обикновено имат ограничена памет, не особено мощен процесор, не мога да изпълняват функциите на L3-L7 така ефективно. Многослойните комутатори все още са най-добри само на 2-ри слой. При тях основното предимство е големият брой портове и по-ниска цена спрямо маршрутизаторите

Методът за предаване на пакети, наричан «виртуален канал», се състои във формирането на единен «виртуален» канал по време на взаимодействието на абонатите за предаването на всички пакети на съобщението. Този метод се реализира с използването на предварително установяване на съединението между взаимодействащите абонати, в процеса на което се формира най-рационален единствен за всички пакети маршрут, по който, за разлика от дейтаграмния метод, всички пакети на съобщението се предават в естествената си последователност.

Виртуалният канал, както и реалният физически канал при комутация на канали, съществува само в течение на сеанса на връзката, при това ресурсите на реалните канали за връзка (пропускателна способност) и на възлите в мрежата (буферна памет), намиращи се по маршрута, се резервират за през цялото време на сеанса.

9. Мрежов слой

Основната идея за въвеждането на мрежовия слой се състои в следното: мрежата в общия случай се разглежда като съвкупност от няколко мрежи и се нарича обединена мрежа или съставна мрежа (internetwork или internet). Мрежите, влизащи в съставната мрежа, също се наричат мрежи или подмрежи (subnet), съставляващи мрежата или просто мрежи.

Подмрежите се свързат с маршрутизатори. Компоненти на обединената мрежа могат да бъдат както локални, така и глабални мрежи.

Мрежовият слой в TCP/IP стека

TCP/IP протоколният стек (Transmission Control Protocol/Internet Protocol) за разлика от OSI/RM съдържа само 4 слоя: приложен, транспортен, мрежов, мрежов интерфейс). Всичките те в една или друга степен съответстват на слоевете на идеалния модел, т. е. изпълняват подобни функции.

Протоколите от мрежовия интерфейс осигуряват интеграцията в съставната мрежа на другите мрежи. На практика този слой не се регламентира, но се поддържа от

всички популярни стандарти на физическия и каналния слой: за локалните мрежи Ethernet, FDDI и т. н., за глобалните мрежи - X.25, frame relay, PPP, ISDN и т. н.

Мрежовият слой (network layer) също така се нарича и internet. Основната му функция е предаване на данни през съставната мрежа (обединяване, свързване на мрежи).

В стандартния модел за взаимодействие на отворените системи във функциите на мрежовия слой влиза решаването на следните задачи:

- предаване на пакетите между крайните възли в обединените мрежи;
- избор на маршрут за предаване на пакети, най-добрия според някакъв критерий;
- съгласуване на различните протоколи на каналния слой, използвани в отделните подмрежи на една обединена мрежа.

Протоколите на мрежовия слой се реализират във вид на програмни модули и се изпълняват на крайните възли (хостовете), а също така и на междинните възли – маршрутизаторите (наричани още шлюзове, въпреки че не са тъждествени). Функциите на маршрутизаторите могат да се изпълняват както от специализирани устройства, така и от универсални компютри със съответстващо програмно осигуряване.

Мрежовият слой, за разлика от долните два слоя, отговаря за предаването на данните в системите за предаване на данни и управлява маршрутизацията на пакетите – предаване през няколко свързващи канала по една или няколко мрежи. Като това обикновено изисква включването в пакета на мрежовия адрес на получателя.

Мрежовият адрес е специфичен идентификатор за всяка междинна мрежа между източника и приемника на информацията.

Блокът от данни, предавани на мрежовия слой, са нарича пакет (packet).

Мрежовият слой реализира:

- откриване на грешки;
- фрагментиране на пакети;
- управление на потоците с данни.

Освен това, към мрежовия слой се отнасят и протоколите за построяване на маршрутните таблици за рутерите: OSPF, RIP, ES-IS, IS-IS, BGP и др.

Базов за цялата архитектура е мрежовият слой, който реализира концепцията за предаване на пакети в режим без установяване на съединения, т. е. чрез дейтаграмен механизъм. Конкретно интернет слоят осигурява възможност за придвижване на пакетите по мрежата, като се използва този маршрут, който в дадения момент е най-рационален.

Транспортният слой е представен от два начина за организация на комуникацията: с гарантирана (TCP) и негарантирана (UDP) доставка на данните.

Протоколите от приложния слой и съответстващите приложения използват или TCP, или UDP за предаване на информацията между двете страни на мрежовото взаимодействие.

За разлика от протоколите от останалите три слоя, протоколите от приложния слой се занимават с детайлите на конкретното приложение и „не се интересуват“ от начините за предаване на данните по мрежата. Този слой постоянно се разширява, защото се присъединяват към старите, преминали многогодишна експлоатация мрежови услуги от типа на Telnet, FTP, TFTP, DNS, SNMP сравнително нови услуги, като например протоколът за предаване на хипертекстова информация HTTP.

10. Internet Protocol

IP е протокол на мрежовия слой в OSI/RM модела.

В качеството на базов на мрежовия слой следва да се определи IP слоя, отговарящ за адресацията и процеса на маршрутизация в глобалните мрежи.

Основен протокол за TCP/IP стека е IP протокол, който осигурява:

- негарантирана доставка на пакетите, понеже предаваните в мрежата пакети могат да се изгубят, да се дублират, да бъдат доставени с нарушена подредба;
- дейтаграмна доставка без установяване на съединение, т. е. всеки пакет се обработва независимо от останалите, при което последователно изпращаните пакети могат да се разпространяват по различни маршрути в мрежата, да си променят реда при получаване и даже да се изгубят;
- максимално възможната доставка на пакетите в този смисъл, че загубата на пакета се случва само когато IP протоколът не намира никакви физически средства за неговата доставка.

IP протоколът изпраща и обработва всяка една дейтаграма като независима порция от данни, т. е. тя няма нищо общо с останалите дейтаграми в глобалната мрежа Интернет.

След изпращането на дейтаграмите от IP протокол в мрежата, последващите действия с тази дейтаграма не се контролират от IP модула на изпращача.

Заглавие на IPv4 протокол

Форматът на IP дейтаграмата включва следните полета (всеки ред съдържа 32 бита, или 4 байта):

- Version – версия на IP протокола (за IPv4 това поле се попълва с 4, което съответства на двоичния запис във вида 0100 в старшия полубайт на първия октет на дейтаграмата);
- IHL – дължина на заглавието в 32 битови думи (Internet Header Length, 4 бита; за IPv4 и дължина на заглавието в 32 битови думи, равна на 5, така в първия байт на IP дейтаграмата е записано 0x45);
- TOS – тип на услугата (Type of Service, 1 байт);
- TL – пълен размер на IP дейтаграмата в байтове (Total Length, 2 байта);
- Identification – 16 битово число, еднакво за всички фрагменти на една дейтаграма (2 байта);
- Flags – флагове (DF – да не се фрагментира, MF – има следващи фрагменти, 3 бита, един флаг е резервиран);
- Fragment Offset – отместване на фрагмента вътре в дейтаграмата в 8 байтови единици (13 бита);
- TTL – време на живот на IP дейтаграмата (Time To Live, стойността на това поле се намалява с 1 при преминаване през поредния маршрутизатор, 1 байт);
- Protocol – по-горе лежащия протокол (например 0x06 за TCP, 0x11 за UDP, 0x01 за ICMP, 1 байт);
- Header Checksum – контролна сума на заглавието на IP дейтаграмата (2 байта);
- Source Address – IP адрес на източника (4 байта);
- Destination Address – IP адрес на получателя (4 байта);
- Options – опции (ако има);
- Padding – байтове за изравняване на полето опции до 32 битова дума;
- Data – данните от по-горе лежащия протокол.

Трите младши бита «Precedence» на TOS определят приоритета на дейтаграмата:

- 111 управление от мрежата;
- 110 междумрежово управление;
- 101 критично;
- 100 повече от мигновено;
- 011 мигновено;
- 010 незабавно;
- 001 спешно;
- 000 обичайно.

Битовете D, T, R, C (следващите 4 бита на TOS) определят желания тип на маршрутизацията:

- D (Delay) – избор на маршрут с минимално забавяне;
- T (Throughput) – избор на маршрут с максимална пропускателна способност;
- R (Reliability) – избор на маршрут с максимална надеждност;
- C (Cost) – избор на маршрут с минимална стойност.

В дейтаграмата може да бъде вдигнат само един от битовете D, T, R, C. Старшият бит на байта TOS не се използва.

Реалното отчитане на приоритетите и избора на маршрута в съответствие със стойността на байта TOS зависи от маршрутизатора, от неговото програмно осигуряване и от настройките му. Маршрутизаторът може да поддържа изчисляване на маршрутите

за всички типове TOS, за някаква част от тях или да игнорира TOS напълно. Маршрутизаторът може да отчита стойността на приоритета при обработването на всички дейтаграми или при обработването на дейтаграмите, изходящи (изпратени) само от някакво ограничено множество от възли в мрежата, или изобщо да игнорира приоритета.

Повечето опции към настоящия момент не се използват. Опциите «идентификатор на потока» (Stream ID) и «Безопасност» (security) се прилагат за ограничен кръг от експерименти, функциите на опциите «Запис на маршрута» (RR - Record Route) и « времева щампа» (Internet Timestamp) се изпълняват от програмата traceroute. Определен интерес представляват само опциите «Loose/Strict Source Routing» (маршрутизация от източника), които са предназначени за указване на дейтаграмата на предопределен от изпращача маршрут.

Прилагането на опциите в дейтаграмите забавя тяхната обработка. Понеже повечето дейтаграми не съдържат опции, т. е. имат фиксирана дължина на заглавието, тяхната обработка е максимално оптимизирана именно за този случай. Появяването на опции прекъсва този високо скоростен процес и извиква стандартен универсален модул IP, способен да обработи всякакви стандартни опции, но за сметка на съществени загуби в бързодействието.

Фрагментация на IP пакетите

По пътя на пакета от източника до получателя може да има локални и глобални мрежи от различни технологии с различни допустими размери на полетата за данни на кадрите в каналния слой (Maximum Transfer Unit – MTU). Понеже в Ethernet мрежата могат да се предават кадри, носещи до 1500 байта данни, за X.25 мрежите е характерен размер на полето за данни на кадъра 128 байта, за FDDI могат да се предават кадри с размер 4500 байта, или всяка мрежова технология си има своите ограничения.

IP протоколът умее да предава дейтаграми, дължината на които е с по-голямо MTU, от това на междинната мрежа, за сметка на фрагментирането – разбиване на „голям“ пакет на някакъв брой части (фрагменти), като размерът на всяка част удовлетворява междинната мрежа. След като всички фрагменти се предадат през междинната мрежа, те ще се събират на възела получател от модула на IP протокола обратно в „голям“ пакет. Събирането (възстановяването на първоначално изпратения пакет) от фрагментите осъществява само получателят, а не някой междинен маршрутизатор. Маршрутизаторите могат само да фрагментират пакети. Също така различните фрагменти на един пакет могат да преминат по различни маршрути.

За да се разпознаят фрагментите на кой пакет принадлежат, се използва поле за идентификация, стойността на което трябва да е еднакво за всички фрагменти на един пакет и да не се повтаря за различните пакети, докато при двата пакета не изтече времето на живот. При делението на данните на пакета, размерът на всички фрагменти, освен на последния, трябва да е кратен на 8 байта. Това позволява да се отдели по-малко място в заглавието за полето отместване на фрагмента.

Вторият бит на полето флагове (More fragments), ако е равен на единица, говори, че даденият фрагмент – не е последен в пакета. Ако пакетът се изпраща без фрагментация, тогава флагът “More fragments” се установява в 0, а полето отместване на фрагмента се запълва с нулеви битове.

Ако първият бит на полето флагове (Don’t fragment) е равен на единица, тогава фрагментацията на пакета е забранена. Ако този пакет трябва да бъде предаден през мрежа с по-малка стойност на MTU, тогава маршрутизаторът ще бъде принуден да го отхвърли (и да съобщи за това на изпращача посредством ICMP протокол). Този флаг се използва в случай, че на изпращача му е известно, че получателят няма достатъчно ресурси, за да възстанови пакета от фрагментите.

Класове IP адреси

Всички IP адреси могат да се разделят на две логические части — номер на мрежата и номер на възела на мрежата (номер на хоста). За да се определи каква част от IP адреса принадлежи към номера на мрежата, а каква - към номера на хоста, се анализират стойностите на първите битове на адреса. Също така, първите битове на IP адреса се използват, за да се определи към кой клас се отнася даденият IP адрес.

Ако първият бит на адреса е 0, тогава мрежата е от клас A и номерът на мрежата е с дължина един байт, останалите 3 байта се интерпретират като номер на възела в мрежата. Мрежите от клас A имат номера в диапазона от 1 до 126. (Номер 0 не се използва, а номер 127 е резервиран за специални цели.) Мрежите от клас A са 126, като броят на възлите в тях може да достига 2^{24} — 2, т. е. 16 777 214 броя.

Ако първите два бита на адреса са равни на 10, тогава мрежата е от клас B. При мрежите от клас B за номер на мрежата и за номер на възела се използват по 16 бита или по 2 байта. Така мрежите от клас B са 16382 на брой с максимален брой възли 2^{16}— 2, или 65 534 броя.

Ако адресът започва с битовата последователност 110, тогава мрежата е от клас C. В този случай за номер на мрежата се използват 24 бита, а за номер на възела - 8 бита. Мрежите от този клас са най-много разпространени, като броят на възлите в тях е ограничен до 254 броя.

Ако адресът започва с битовата последователност 1110, тогава той е от клас D и е групов адрес - multicast. Ако в пакета в качеството на адрес на получателя е посочен адрес от клас D, тогава този пакет трябва да бъде получен от всички възли, на които е присвоен дадения адрес.

Ако адресът започва с битовата последователност 11110, тогава даденият адрес е от клас E. Адресите от този клас са резервирани за бъдеща употреба.

Използване на маските в IP адресацията

Традиционната схема на разделяне на IP адреса на номер на мрежата (NetID) и номер на възела (HostID) е основана на понятието клас, който се определя от стойността на няколко водещи битове в адреса.

Възможно ли е използването на някакъв друг признак, с помощта на който да може по-гъвкаво да се определи границата между номера на мрежата и номера на възела? В качеството на такъв признак е получила широко разпространение маската.

Маската е число, което се използва заедно с IP адреса. Двоичното представяне на маската съдържа единици само в тези разряди, които следва да се интерпретират в IP адреса като номер на мрежата. Понеже номерът на мрежата е съставяща част от адреса, единиците в маската трябва да представляват непрекъсната последователност.

Получава се така, че за стандартните класове на мрежите, маските имат следните стойности:

- клас A - 11111111. 00000000. 00000000. 00000000 (255.0.0.0);
- клас B - 11111111. 11111111. 00000000. 00000000 (255.255.0.0);
- клас C - 11111111. 11111111. 11111111. 00000000 (255.255.255.0).

Като се снабди всеки IP адрес с маска, можем да се откажем от понятието класове адреси и да се реализира значително по-гъвкава система за адресация.

Съществува също така съкратен вариант за записването на маската, наричан префикс.

Например за мрежа 80.255.147.32 с маска 255.255.255.252, може да се използва запис от вида 80.255.147.32/30, където „/30“ задават броя на двоичните единици в маската, т. е. тридесет бинарни единици (отброяването се провежда отляво надясно).

Ефективно средство за структуризация на IP мрежи представляват мрежовите маски. Маските позволяват да са раздели една мрежа на няколко подмрежи. Маските с еднаква дължина се използват за разделяне на мрежите на подмрежи с еднакъв размер, а маските с променлива дължина - за разделене на мрежата на подмрежи с различен размер. Използването на маските модифицира алгоритъма на маршрутизацията, затова в този случай се предявяват особени изисквания към протоколите за маршрутизация в мрежата, към техническите характеристики на маршрутизаторите и към процедурите за тяхното конфигуриране.

Механизмът на маските е разпространен в IP маршрутизацията, при което маските могат да се използват за най-различни цели. С тяхна помощ администраторът може да структурира своята мрежа, без да изисква от доставчика на услугите допълнителни номера на мрежи. На база на този механизъм доставчиците на услуги могат да обединяват адресните пространства на няколко мрежи като използват префикси с цел намаляване на обема на таблиците за маршрутизация и така да повишат производителността на маршрутизаторите. Също така, при представяне на маската във вид на префикс, записът е значително по-кратък.

Настоящето и бъдещето на IP мрежите е пряко свързано с безкласовата маршрутизация (Classless Inter-Domain Routing – CIDR), която решава две основни задачи. Първата се състои в по-икономичното изразходване на адресното пространство – благодарение на CIDR доставчиците на услуги получават възможност «да нарежат» блокове с различни размери от отделеното им адресно пространство в точно съответствие с изискванията на всеки клиент. Втората задача се изразява в намаляването на броя на записите в маршрутните таблици за сметка на обединяването на маршрутите - един запис в маршрутната таблица може да представлява голям брой мрежи с общ префикс.

Присвояването на IP адреси на хостовете се реализира по един от следните два начина:

- ръчно, като се настройва от системния администратор (и се съгласува с дадения мрежов сегмент);
- автоматично, с използване на специални протоколи (в частност, с помощта на DHCP протокола - Dynamic Host Configuration Protocol, протокол за динамична настройка на хостовете).

Особени IP адреси

В IP протокола има споразумения за особена интерпретация на следните IP адреси:

- 0.0.0.0 - представлява адрес на шлюза по подразбиране (default gateway), т.е. адрес на хоста, към който трябва да се насочват информационните пакети, ако те не са намерили получателя в локалната мрежа (в маршрутната таблица);
- 255.255.255.255 – общодостъпен адрес. Съобщенията, предадени за този адрес, ще бъдат получени от всички възли на локалната мрежа, съдържаща хоста източник на съобщението (то няма да се предаде в други локални мрежи / мрежови сегменти);
- «Номер на мрежата».«всички нули» – адрес на мрежа (например 192.168.23.0);
- «Всички нули».«номер на хост» – възел от дадената мрежа (например 0.0.0.10). Може да се използва за предаване на съобщения за конкретен възел в локалната мрежа;
- Ако в полето номер на хост получател стоят само единици, тогава пакетът, притежаващ такъв адрес, се изпраща на всички възли в мрежата със зададен номер на мрежата. Например пакет с адрес 192.190.21.255 се доставя на всички възли на мрежата 192.190.21.0. Такова предаване на съобщения се нарича общодостъпно (broadcast).

При адресацията е необходимо да се отчитат тези ограничения, които се внасят с особеното предназначение на някои IP адреси. Така, нито номерът на мрежата, нито номерът на възела не могат да се състоят само от двоични единици или само от двоични нули. Следователно максималният брой възли за една мрежа на практика трябва да се намали с две. Например за мрежите от клас C за номер на възел са заделени 8 бита, които позволяват задаване на 256 номера: от 0 до 255. Обаче на практика максималният брой възли в мрежа от клас C не може да превишава 254, понеже адресите 0 и 255 имат специално предназначение.

Особен смисъл има IP адреса, първият октет на който е равен на 127. Той се използва за тестване на програми и взаимодействие на процесите в рамките на една машина. Когато програмата изпраща данни за IP адрес 127.0.0.1, тогава се образува «примка» (loop). Данните не се предават в мрежата, а се връщат на модулите на по-горния слой в качеството на току-що получени. Затова в IP мрежите се забранява присвояването на хостовете на IP адреси, започващи с 127. Този адрес има наименование loopback. Може този адрес 127.0.0.0 да се отнесе към вътрешна мрежа за модула за маршрутизация на възела, а адреса 127.0.0.1 – към адрес на този модул за вътрешната мрежа.

В IP протокола няма понятие броудкаст в този смисъл, в който то се използва в протоколите на каналния слой за локалните мрежи, когато пакетите трябва да бъдат доставени до абсолютно всички възли. Както ограниченият броудкаст IP адрес, така и броудкаст IP адресът имат граници на разпространение в обединената мрежа – те са ограничени или от мрежата, в която се намира възелът, източник на пакета, или от мрежата, номерът на която е посочен в адреса на получателя. Затова разделянето на обединената мрежа с помощта на маршрутизатори на части локализира броудкастно заливане с пакети в рамките на една от съставляващите обединената мрежа. Това става лесно, защото няма как да се адресира пакет така, че той да е предназначен за всички възли в обединената мрежа.

Частни IP адреси (използвани за локални цели)

Всички използвани адреси в Интернет трябва предварително да се регистрират, като това гарантира тяхната уникалност в мащабите на цялата планета. Такива адреси се наричат реални или публични IP адреси.

За локалните мрежи, които не са свързани с Интернет, регистрация на IP адресите не е наложителна, като на практика могат да се ползват всички възможни адреси. Обаче, за да не се получи конфликт при последващо свързване на такава мрежа към Интернет, се препоръчва използването в локалните мрежи единствено на следните диапазони, така наречените частни (private) IP адреси (за Интернет тези адреси не съществуват и те да бъдат използвани там е невъзможно):

10.0.0.0 – 10.255.255.255

172.16.0.0 – 172.31.255.255

192.168.0.0 – 192.168.255.255

Един от най-сериозните недостатъци на интернет протокола IPv4 е относително неголемият брой възможни адреси - около 4,23 милиарда. Това число вече не изглежда толкова голямо в сравнение с броя на задействаните устройства, свързани към мрежата Интернет. И до ден днешен използването на IPv4 преминава в щатен режим, понеже се прилагат различни технологии за икономия на мрежовите адреси, в частност и технологията NAT (Network Address Translation, преобразуване на мрежовите адреси). Но вече става ясно, че дните на IPv4 изтичат, понеже в най-близко бъдеще се предвижда възможността за достъп до Интернет на всички битови прибори (хладилници, печки и

т.н.), за реализация на управление на дадените прибори отдалечено, посредством мрежата от всяка една точка на Земята.

В получилата се ситуация преходът към нов формат на мрежов адрес става неизбежен. Групата за проектиране на Интернет IETF през 1990 г. започва да създава новата версия на мрежовия протокол - IPv6.

Частните адреси отслабват проблема с недостиг на публични IPv4 адреси.

Транслаторът на мрежови адреси NAT привежда един частен адрес в един публичен. Той транслира вътрешен (inside) локален (частен) IP адрес на клиента във външен глобален (публичен) адрес.

Транслаторът PAT (NAT Overload) комбинира един общодостъпен IP адрес с набор от номера на порта на възела на източника, т.е. формира съвкупност от комплексни адреси, наричани сокети.

NAT повишава безопасността на мрежата, понеже скрива вътрешните (частните) IP адреси от външните мрежи.

Статичният NAT осигурява постоянни двойки от локален и глобален адрес.

В случая за динамичен NAT вътрешните локални адреси се преобразуват в глобални адреси (обикновено публични), които се вземат от набор (пул) от адреси.

За да се конфигурира динамичен NAT, на маршрутизатора е необходимо да се установи списък за достъп ACL, разрешаващ транслацията на подлежащите за превод адреси и пула от глобални адреси.

Технологията за преадресация на порта позволява на клиентите от външните мрежи да получат достъп до възлите (сървърите) във вътрешна мрежа чрез специално конфигуриран транслатор на адресите - NAT.

През периода на преход от IPv4 мрежи към IPv6 мрежи се използва технологията за транслация на адресите NAT64.

ICMP протокол (Internet Control Message Protocol)

ICMP (протокол за управляващи съобщения в Интернет мрежата) се инкапсулира в IP дейтаграма (при което полето Protocol в IP заглавието съдържа 0x01).

- Type – тип на ICMP съобщението (1 байт);
- Code – код на съобщението (зависи от стойността на полето Type, 1 байт);
- Checksum – контролна сума (2 байта);
- Data – данни.

ARP и RARP протоколи (Address Resolution Protocol и Reversed ARP)

ARP/RARP протоколът, заглавията и данните, на който се инкапсулират непосредствено в кадри на каналния слой, а не в IP дейтаграма, е предназначен за взаимосвързване на адресациите на каналния (най-често MAC адресация) и мрежовия слой (IP адресация).

Установяването на съответствие между IP адреса и апаратния адрес (наричан още хардуерен, физически, MAC адрес) се реализира от протокола за разрешаване на адресите ARP, който за тази цел преглежда ARP таблиците. Ако търсеният адрес отсъства, тогава се изпълнява броудкастна ARP заявка.

Протоколите ARP (разпознаване на MAC адреса по зададен IP адрес) и RARP (откриване на IP адрес по известен MAC адрес) свързват каналния и мрежовия слой в OSI/RM модела. И двата протокола носят броудкастен характер, затова се използват само в локални мрежи (в рамките на един сегмент на мрежовата архитектура на каналния слой, което означава "до първия маршрутизатор").

Необходимо е да се обърне внимание, че ARP и RARP организират взаимната връзка не само за Ethernet и IP, но и за другите мрежови архитектури и протоколи на мрежовия слой. Понеже Етернет е станал лидиращ протокол в локалните мрежи, затова най-често с него се свързва.

Форматът на ARP и RARP пакетите е следния:

- Hardware Type – тип на мрежовата архитектура (канален слой, 0x0001 за 10 Mb Ethernet);
- Protocol Type – идентификатор на протокола на мрежовия слой (0×0800 за IP);
- HW Address Length – дължина на адреса на каналния слой (0×06 за MAC адреса);
- Protocol Address Length – дължина на адреса на мрежовия слой (0×04 за IP адреса);
- Opcode – код на операцията (1 – ARP заявка, 2 – ARP отговор, 3 – RARP- заявка, 4 – RARP отговор);
- Sender и Target HW Address и Protocol Address – адреси на каналния и мрежовия слой на източника и получателя съответно.

Част от полетата може да са празни. Така, при формирането на ARP заявка, полето Target HW Address остава непопълнено. Станцията, притежаваща информация за търсения адрес, ще сформира ARP отговор, в който ще подаде търсения адрес на каналния слой.

Непосредствено връзката между IP адреса и MAC адреса се осъществява с помощта на така наречената ARP таблица, където на всеки ред се посочва съответствието на IP адреса и MAC адреса.

В ARP таблицата, освен IP и MAC адреса, още се посочва типа на връзката, като съществуват два типа записи:

- Статичните записи се създават ръчно. Те съществуват до тогава, докато компютърът или маршрутизаторът остават включени.

- Динамичните записи трябва периодично да се обновяват. Ако запис не се обнови в течение на определено време (например около 2 минути), тогава той се изтрива от таблицата.

В ARP таблицата се съдържат записи не за всички възли в мрежата, а само за тези, които активно участват в мрежовите операции. Такъв метод за съхраняване се нарича ARP кеш.

DHCP

Dynamic Host Configuration Protocol (DHCP - протокол за динамично конфигуриране на възлите) позволява да се автоматизира процеса на задаване на IP адреси на работните станции от диапазона (пула) адреси, отделен от провайдера на администратора.

Получаването на адресите може да се реализира по три начина: ръчно, автоматично и динамично.

В ръчен режим администраторът сам задава на устройството определен от DHCP протокола статичен IP адрес.

В автоматичен режим DHCP протоколът отдава на устройството за постоянно ползване статичен IP адрес от пула с адреси.

В динамичен режим DHCP протоколът отдава от пула с адреси на устройството IP адрес за определен период от време в аренда.

В локалните мрежи сървърът се конфигурира или на отделен персонален компютър, или на локален маршрутизатор, който получава IP адреса си от DHCP сървъра на провайдера.

Клиентът, на когото е необходим адрес, изпраща в мрежата броудкастна заявка за откриване на DHCP сървърите DISCOVER с MAC адрес на получателя FF-FF-FF-FF-FF-FF.

Сървърът отговаря с оферта за арендоване на IP адрес (DHCP OFFER) с използване на едноадресен получател.

В локалната мрежа може да има няколко DHCP сървъра. Затова клиентът първо си избира сървър и тогава му изпраща броудкаст заявка DHCP REQUEST за получаване на IP адрес. Заявката е броудкастна, за да научат другите DHCP сървъри, че тяхната оферта е отклонена.

Сървърът отговаря положително с потвърждение DHCP PACK в едноадресен режим или дава отрицателен отговор с DHCP NAK.

Удължаването на срока на арендата на IP адреса става като се изпраща DHCP заявка REQUEST в едноадресен режим.

При конфигурирането на DHCP сървър на маршрутизатора се задава пул от адреси, освен адресите, които за запазени за сървърите, маршрутизаторите и мрежовите принтери.

Маршрутизаторът също така може не само да бъде DHCP сървър, но и да бъде клиент, когато получава IP адрес за някой свой интерфейс от сървъра на провайдера.

В IPv6 мрежите индивидуалните глобални адреси могат да бъдат назначавани автоматично по различен начин:

- Използването само на обявленията на маршрутизатора, когато се реализира автоматично получаване на адресната информация без проследяване на състоянието - SLAAC.
- Използване на обявленията на маршрутизатора и DHCPv6 протокол без проследяване на състоянието (DHCPv6 SLAAC);
- Използване на DHCPv6 протокол с проследяване на състоянието.

11. IPv6 протокол

В наши дни IPv6 протокол активно се използва от множество мрежи по целия свят, но все още не е получил толкова широко разпространение в Интернет, както IPv4.

Основни особености на IPv6 протокол:

1. Дължината на IP адреса е увеличена до 16 байта, като така се предоставя на потребителите практически неограничено адресно пространство;
2. Опростена структура на заглавието, съдържащо само 8 полета (вместо 13 в IPv4 протокола). Последното позволява на маршрутизаторите по-бърза обработка на пакетите, т. е. повишава се тяхната производителност;
3. Подобрена поддръжка на незадължителни параметри, понеже в новото заглавие изискваните по-рано полета стават незадължителни, а измененият начин за представяне на незадължителните параметри ускорява обработването на пакетите от маршрутизаторите за сметка на отсъствието на обработване на тези параметри;
4. Подобрена система за безопасност. Автентификацията и конфиденциалността са основни характеристики на новия IP протокол;
5. Предвидена е възможност за разширявяна на типовете (класовете) на предоставяните услуги, които могат да се появят в резултат на очаквания растеж на мултимедийния трафик. Отделено е по-голямо внимание на типа на предоставяните услуги. За тази цел в заглавието на пакета на IPv4 е било предвидено 8 разрядно поле.

Заглавие на IPv6 протокол

- Версия — за IPv6 стойността на това поле трябва да е 6.
- Клас на трафика – използва се, когато трябва да се различават пакетите с различни изисквания към доставката в реално време.
- Етикет на потока – прилага се с цел установяване между изпращача и получателя на псевдосъединение с определени свойства и изисквания. Например потокът от пакети между два процеса на различни хостове може да притежава строги изисквания към забавянията, което ще наложи резервиране на пропускателна способност.
- Дължина на полезните данни – съобщава, колко байта следват след 40 байтовото заглавие.
- Следващо заглавие – съобщава, кое от допълнителните заглавия следва след основното.
- Максимален брой на транзитните възли – аналог на време на живот (TTL).
- Допълнителни заглавия:
    - Параметри на маршрутизацията – разнообразна информация за маршрутизаторите;
    - Параметри за получаване – допълнителна информация за получателя;
    - Маршрутизация – частичен списък на транзитните маршрутизатори по пътя на пакета;
    - Фрагментация – управление на фрагментите на дейтаграмите;
    - Автентификация (IPsec AH) – проверка за достоверност на изпращача;
    - Шифровани данни (IPsec ESP) – информация за шифровано съдържимо.

Структура на IPv6 пакета

Структурата на IPv6 пакета значително се различава от тази на IPv4 пакета. Това се проявява най-силно във възможността на присъствие на няколко заглавия. Освен основното заглавие, което винаги е налично, пакетът може да има няколко допълнителни заглавия, които могат да съдържат информация, необходима за качественото предаване на пакета.

Дължината на заглавието е точно 40 байта. Като правило, структурата на заглавието на IPv6 е по-проста от тази на IPv4.

В заглавието не се задават никакви допълнителни параметри. Вместо тях IPv6 добавя допълнителни заглавия. Такива заглавия могат да съдържат информация за AH и ESP (както и в IPv4), а също и информация за преминаването през транзитните възли, за маршрута, за фрагментацията и за получателя. В настоящия момент IPv6 поддържа няколко заглавия за разширения.

Като допълнителни заглавия могат да се използват следващите варианти:

- заглавие за маршрутизация, съдържащо пълния маршрут, в случая за маршрутизация от източника (IPsrc);

- заглавие за фрагментация, съдържащо информация за фрагментацията на изходния IP пакет;
- заглавие за автентификация, съдържащо информация, необходима за автентификацията на крайните възли и осигуряването на цялостност на съдържимото на IP пакетите;

- заглавие на системата за безопасност, съдържащо информация, необходима за осигуряването на конфиденциалност на предаваните данни чрез шифроване на пакетите;

- специални параметри, необходими за обработката на пакетите в процеса на предаването им по мрежата;
- параметри за получателя, съдържащи допълнителна информация за хоста, получател на пакета.

В IPv6 пакетът може да се разбие на фрагменти само при изпращача си. Събирането на пакета може да се изпълни само на възела на получателя. Тогава се прилага допълнителното заглавие с разширение за фрагментация.

В IPv6 минималният размер на MTU е 1280 байта. Следователно пакетите на IPv6, размерът на които е по-малък от това ограничение, не се разбиват на фрагменти. За предаване на IPv6 пакети по свързващите линии с размер на MTU по-малък от 1280 байта, тези пакети трябва да се разбиват и събират на нивото на свързващия канал.

IGMP (Internet Group Management Protocol — протокол за управление на групите в Интернет) протокол е заменен с MLD протокол. MLD протокол изпълнява същите функции, както и IGMP протокол в IPv4. Той използва ICMPv6 протокол, в който са предвидени няколко нови типове, предназначени за MLD.

В IPv6 са вградени функциите на ARP протокол. Те са реализирани в алгоритмите за автоматична настройка на адресите и търсене на съседи, които използват ICMPv6 протокол. Във връзка с това ARP6 протокол не е разработван.

Такава структура на IPv6 пакета осигурява следните предимства:

- по-ниско натоварване за маршрутизаторите, понеже всички допълнителни заглавия се обработват само в крайните възли;
- по-висока функционалност и отвореност за внедряване на нови механизми в IP протокола поради използването на голям брой допълнителни параметри.

Система за адресация

Главната цел при промяната на системата за адресация не е механичното увеличаване на адресното пространство, а повишаването на ефективността на работа на TCP/IP стека като цяло.

Обикновено първите 64 бита задават номера на мрежата, а вторите 64 бита – номера на хоста. Често пъти в качеството на номер на хост или на негов компонент IPv6 адресът се получава на база на MAC адреса или на друг идентификатор на интерфейса.

В подмрежите с някои префикси на IPv6 архитектурата е по-сложно от IPv4 архитектурата.

Броят на IPv6 адресите е 79 228 162 514 264 337 593 543 950 336. Това е 1028 пъти по-голямо число от броя на IPv4 адресите.

Реалните и частните адреси в IPv4 се наричат още външни и вътрешни адреси. В IPv6 се използва аналогична структура на адресите, но с някои съществени разлики. Адресите се делят на външни и временни (временните адреси преди са се наричали анонимни). За разлика от вътрешните адреси в IPv4, временните се разпознават в глобалната мрежа. Те се използват с друга цел. Временният адрес скрива идентификатора на клиента, установяващ съединението (от съображения за защита). Срокът на действие на временният адрес е ограничен. Такъв адрес не съдържа идентификатор на интерфейса, т. е. MAC адрес. Като правило, временният адрес не може да се различи от обикновения външен адрес.

Видове адреси

Unicast - Идентификатор за единичен интерфейс. Пакет, изпратен към такъв адрес, се доставя на интерфейса, посочен в адреса.

- Идентифицира конкретен интерфейс в мрежата.
- Пакет, изпратен за такъв адрес, ще бъде доставен на този интерфейс.
- В IPv6 има няколко типа unicast адреси:

    - Global unicast
    - Link-local (FE80::/10)
    - Unique local unicast (FC00::/7)

Anycast - Идентификатор на набор от интерфейси (принадлежащи на различни възли). Пакетът, изпратен към такъв адрес, се доставя на един от интерфейсите, посочен в адреса (най-близкия, в съответствие с цената, определена от протокола за маршрутизация).

- Присвояват се на група от интерфейси, които обикновено принадлежат на различни устройства.
- Пакет, изпратен за anycast адрес, се доставя на един участник от групата от интерфейси, обикновено на най-близкия от гледна точка на маршрутизатора (маршрутизаторът използва метрика на протоколите за маршрутизация, за да определи най-близкия интерфейс).
- Форматите за unicast и anycast адреси са еднакви.
- Адресите anycast се използват от маршрутизаторите.
- Anycast адресът не бива да се използва в качеството на адрес на изпращача в IPv6 пакета.

Multicast - Идентификатор на набор от интерфейси (обикновено принадлежащи на различни възли). Изпратеният към групов адрес пакет се доставя на всички интерфейси, зададени с този адрес.

- Идентифицират група от интерфейси, които обикновено принадлежат на различни устройства.
- Пакет, изпратен за такъв адрес, ще бъде доставен на всички интерфейси от групата.
- За multicast адресите префиксът е FF00::/8.
- В IPv6 няма броудкаст адреси, а тяхната функция е предадена на груповите адреси.

Модел на адресацията

IPv6 адресът (от всички типове) се асоциира с интерфейсите, а не с възлите. Също така всеки интерфейс принадлежи точно на един възел, а уникастният адрес на интерфейса може да идентифицира възела.

IPv6 уникастният адрес се съотнася само с един интерфейс. На един интерфейс могат да съответстват много IPv6 адреси от различен тип (unicast, anycast и multycast). Съществуват две изключения от това правило:

- Единичен адрес може да се задава на няколко физически интерфейса, ако приложението ги разглежда като единно цяло при представянето му в Интернет.
- Маршрутизаторите могат да притежават интерфейси, на които не е присвоен никакъв IPv6 адрес, за съединения от типа точка-точка, за да се изключи необходимостта ръчно да се конфигурират и обявяват тези адреси. Адресите не са необходими за съединения точка-точка на маршрутизаторите, ако тези интерфейси не се използват в качеството на точка за източник или получател при изпращане на IPv6 дейтаграми. Маршрутизацията тук се реализира според схема, близка до използваната от CIDR в IPv4.

IPv6 съответства на модела за IPv4, където подмрежата се асоциира с канал. На един канал могат да съответстват няколко подмрежи.

Формати за представяне на IPv6 адреси

Формат с шестнадесетични числа и двоеточия

Този формат е най-предпочитан и той има следния вид: n:n:n:n:n:n:n:n. Тук всеки знак n съответства на 4 значно шестнадесетично число (общо 8 шестнадесетични числа, за всяко число са заделят 16 бита).

Например: 1FA9:FFFF:2621:ACDA:2245:BF98:3412:4167.

Съкратена форма

Поради голямата си дължина адресът обикновено съдържа много поредни нули. За опростяване на записа на адресите се използва съкратена форма, в която съседни последователности от нулеви блокове се заменят от двойка символи двоеточие (::). Обаче, такъв символ може да се среща в адреса еднократно.

Например:

- Адресът за групово разпращане FFEA:0:0:0:0:CA28:1210:4362 има следната съкратена форма: FFEA::CA28:1210:4362.
- Адресът за едноадресно разпращане 3FFE:FFFF:0:0:8:800:02A1:0 в съкратена форма има следния вид: 3FFE:FFFF::8:800:02A1:0.
- Loopback адресът 0:0:0:0:0:0:0:1 в съкратена форма е ::1.
- Неопределеният адрес 0:0:0:0:0:0:0:0 се преобразува в :: .

Смесена форма

Тази форма представлява съчетание на адреси на протоколите IPv4 и IPv6. В този случай адресът има формата n:n:n:n:n:n:d.d.d.d, където всеки символ n съответства на 4 значно шестнадесетично число (6 шестнадесетични числа, като за всяко число се заделят 16 бита), а d.d.d.d – част от адреса, записана във формата на IPv4 (32 бита).

Например:

- 0:0:0:0:0:0:19.8.62.32
- 0:0:0:0:0:FFFF:111.214.2.34
- или в съкратен вид:
  - ::73.3.68.45
  - ::F2F3:129.131.32.31

На интерфейс от IPv6:

- Могат да бъдат зададени няколко адреса от различни типове.
- На всеки интерфейс трябва да има поне един loopback (::1/128) и един link-local адрес.

All Nodes

All Nodes адресът идентифицира групата на всички IPv6 хостове, с граница 1 (interface-local) или 2 (link-local):

FF01:0:0:0:0:0:0:1

FF02:0:0:0:0:0:0:1

All Routers

All Routers адресите идентифицират групата на всички IPv6 маршрутизатори, с граница 1 (interface-local), 2 (link-local) или 5 (site-local):

FF01:0:0:0:0:0:0:2

FF02:0:0:0:0:0:0:2

FF05:0:0:0:0:0:0:2

Задължителни адреси за хост

Хостът е длъжен да разпознава следните адреси, в качеството на адреси, които го идентифицират:

- Link-local адрес за всеки интерфейс;
- Всеки допълнителен unicast или anycast адрес, който е бил настроен на интерфейс на хоста (ръчно или автоматично);
- loopback адрес;
- All-Nodes multicast адрес;
- Solicited-Node multicast адрес за всеки един от настроените unicast или anycast адреси;
- Multicast адреси за всякакви други групи, към които принадлежи хоста.

Задължителни адреси за маршрутизатора

Маршрутизаторът е длъжен да разпознава всички адреси, които е длъжен да разпознава и хоста, плюс следните, в качеството си на адреси, които го идентифицират:

- Subnet-Router Anycast адрес за всички интерфейси, за които той е настроен за работа в качеството на маршрутизатор;
- Всякакви други Anycast адреси, които са настроени на маршрутизатора;
- All-Routers multicast адрес.

Специални адреси

Някои адреси (например изброените в rfc5156 и в списъка на IANA) имат специално предназначение и тяхното използване трябва да бъде обосновано.

- 2001:7f8::/32 - за раздаване на блокове на точките за обмен на интернет трафик (ripe-510);
- 2001:678::/29 - за раздаване на /48 коренови услуги (ripe-510);
- 2001:0::/32 - клиенти на Teredo (rfc4380);
- 2001:db8::/32 – за документация и примери (rfc3849);
- 2002::/16 - реализация на 6to4 (rfc3056);
- 64:ff9b::/96 - реализация на NAT64 (rfc6052);
- ::FFFF:0:0/96 – изобразяване на IPv4 адреси (rfc5156);

Протокол за откриване на съседи

Протоколът за откриване на съседи (Neighbor Discovery, ND) е отговорен за автоматичното настройване на адреса на крайните и на междинните възли в мрежата, откриването на други възли по линията, определяне на адресите на другите възли на

каналния слой (вместо ARP, който се използва за IPv4), откриване на конфликти при адресирането, търсене на достъпни маршрутизатори и DNS сървъри, определене на префикса на адреса и поддръжката на достъпност и на информацията за пътищата към други активни съседни възли.

Този протокол установява пет различни типа ICMPv6 пакети за реализация на функциите на IPv6, сходни с ARP, ICMP, IRDP и Router Redirect на протоколите за IPv4. Също така той предоставя множество подобрения относно IPv4 аналозите.

NDP установява следните пет типа ICMPv6 пакети:

1. Заявка за достъпност на маршрутизаторите;
2. Отговор на маршрутизатора;
3. Заявка за достъпни съседи;
4. Отговор на съседа;
5. Пренасочване.

Тези съобщения се използват за осигуряването на следните функционалности:

- *Откриване на маршрутизатора (Router Discovery)* — хостовете могат да откриват маршрутизатори, които се намират на свързаната линия;
- *Откриване на префикса (Prefix Discovery)* — хостовете откриват префикси, които определят кои получатели се намират на тяхната линия (хостовете използват префиксите, за да определят кои получатели са достъпни директно, а кои са достъпни само с използване на маршрутизатор);
- *Откриване на параметри (Parameter Discovery)* — хостовете получават параметри на връзката (например MTU);
- *Автоматична настройка на адреса (Address Autoconfiguration)* — хостовете автоматично настройват адреса на интерфейса;
- *Преобразуване на адреса (Address resolution)* — хостовете определят адреса на каналния слой на съседите според IP адреса на получателя;
- *Определяне на next-hop (Next-hop determination)* — алгоритъм за установяване на съответствието между IP адреса на получателя и IP адреса на съседа, на когото трябва да се изпрати трафика, за да бъде доставен на получателя. Next-hop може да бъде маршрутизатор или самия получател;
- *Определяне на недостижимост на съседа (Neighbor Unreachability Detection, NUD)* — хостовете определят, че съседът вече е недостъпен;
- *Определяне на дублиран адрес (Duplicate Address Detection, DAD)* — хостът определя, че адреса, който той иска да използва, не се ползва от други хостове;
- *Пренасочване (Redirect)* — маршрутизаторът уведомява хоста, че има по-добър маршрутизатор (first-hop), за изпращането на трафик за конкретен получател.

Рекурсивен DNS сървър (RDNSS) и списък за търсене DNS (DNSSL) са нова функционалност, която не се поддържа винаги от програмното осигуряване.

Сравнение на мобилния IPv4 и мобилния IPv6

Мобилният IPv6 има много общи свойства с мобилния IPv4, но той е интегриран в IPv6 и предлага много други нови подобрения.

Тук отсътства необходимостта, както в мобилния IPv4, да се прилагат специфични маршрутизатори в качеството на «външни агенти». Мобилният IPv6 работи навсякъде без някаква специална поддръжка, изисквана от локалния маршрутизатор.

Поддръжката за оптимизация на маршрутите е основна част от протокола, а не е нестандартен набор от разширения.

Оптимизацията на маршрутите при мобилния IPv6 може да работи надеждно даже без предварително организирани контексти за безопасност. Предполага се, че оптимизацията на маршрутите може да бъде разгърната в глобален мащаб между всички мобилни възли и възли кореспонденти.

В мобилния IPv6 е интегрирана поддръжка, предоставяща възможност за разумна съвместна работа за оптимизация на маршрутите с маршрутизатори, които реализират «входяща» филтрация.

Механизмът за определяне на недостижимостта на съседите при IPv6 гарантира симетрична достижимост между мобилния възел и неговия подразбиращ се маршрутизатор в текущото местоположение.

В мобилния IPv6 повечето пакети, изпращани към мобилния възел, когато той се намира извън дома, се изпращат с помощта на заглавието за маршрутизация на IPv6, а не с помощта на IP инкапсулация, като така се съкращават допълнителни разходи в сравнения с мобилния IPv4.

Мобилният IPv6 е отделен от всеки конкретен канален слой, понеже той вместо ARP използва IPv6 Neighbor Discovery протокол. Това подобрява устойчивостта на протокола.

В мобилния IPv6 се използва IPv6 инкапсулация, което отстранява необходимостта да се управлява допълнително състоянието на тунела.

В мобилния IPv6 механизмът за динамично определяне на адреса на домашния агент връща на мобилния възел единичен отговор. Управляемият броудкастен подход, използван в IPv4, връща отделни отговори от всеки домашен агент.

IPsec

IPsec е протокол за тунелиране на мрежовия трафик на 3 слой, предназначен за безопасно предаване на IP пакети през мрежа за общо използване. Протоколът IPsec предполага два съществено различни режима на работа:

- Тунелният режим е предназначен за съединение на две частни IP подмрежи през мрежа за общо използване. Пакетите на частните мрежи се инкапсулират в нови IP пакети, заедно със заглавието (включвайки IP адреса на частните мрежи). Този режим е предназначен за мрежови устройства (IPsec шлюзове), на които, като правило, не се изпълняват приложни програми. Приложните хостове се разполагат в частните мрежи, които са защитени от тези шлюзове.
- Транспортният режим е предназначен за съединение на два приложни процеса (на хостове), непосредствено свързани към мрежа за общо използване. В този режим се скрива само съдържимото на IP пакета, заглавието на пакета не се променя.
- Следва да се отбележи, че въпреки стандартизацията на IPsec, не всички съществуващи реализации са напълно съвместими. И колкото стандартите и спецификациите са по-обширни, толкова повече детайли в тях допускат неоднозначно тълкуване.
- Security Association (SA) се нарича съединението, което предоставя услуга за осигуряване на безопасен трафик, който ще се предава през него.
- За работата на тунела са необходими, в общия случай, два ключа: за автентификация и за защита на трафика. Тези ключове могат да се задават по два начина, които са взаимосвързани с факта на съществуване на тунела:
  - *Постоянно съществуващ тунел*. Създава се незабавно след образуването на SA и съществува до тогава, докато не се изтрие от конфигурацията на устройството. Всички параметри на тунела, като ключове и алгоритми за защита на трафика, се определят от администратора на устройството ръчно, статично, и трябва да бъдат установени еднакво за двата края (на двете страни). Никакви начини за автоматично прекратяване на тунела или изменение на неговите параметри в процеса на работата му не са предвидени.
  - *Динамично създаван тунел*. За създаването на такъв тунел се използва процедура за автоматично съгласуване на ключовете, определена от протокола IKE (Internet Key Exchange). В резултат на образуването на SA се подготвят условия за създаването на тунел, но непосредствено създаването на тунела става, като правило, тогава, когато на една от страните се появяват данни за предаване по тунела. Създаването на тунел незабавно след образуването на SA също е възможно, като това се реализира с допълнителни опции. Ключовете, индексът на тунела (SPI) и конкретният алгоритъм за защита на трафика (от списъка с разрешени на едната и на другата страна) се съгласуват и се пресъгласуват автоматично според необходимостта.

Създаденият тунел може да бъде прекратен от всяка една от участващите страни в произволен момент от време. В частност, това може да стане по следните причини:

- Изтичане на установеното време без активност.
- Изтичане на установеното време на живот на тунела (според времето или според обема на предадения трафик).
- Излизане от строя на отдалечен шлюз или загуба на връзката с него, установени с помощта на механизма DPD (Dead Peer Detection).
- Ръчно по инициатива на администратора на единия от VPN шлюзовете, с помощта на съответната команда.
- След прекратяването на тунела, той остава в състояние на готовност и може да бъде автоматично създаден отново по инициатива на една от страните. При това неговите параметри се съгласуват отново.

Постоянните IPsec тунели се използват изключително рядко. Една от причините за това е необходимостта ръчно да се въвеждат дълги ключове. Другата, по-съществената, е в това, че ключовете се задават статично, без ограничение на срока на действието им. Те могат да бъдат разбити за крайно време (поне теоретично), достатъчно неголямо — при съвременните изчислителни мощности и при разпределена атака с помощта на мрежа (ботнети) от зомби компютри. Постоянните тунели също могат да бъдат реализирани с помощта на допълнителни опции.

Технологията IPsec по своята същност не предполага използване на клиент и сървър. Процедурите за автентификация, за съгласуване на параметрите и за установяване на тунел могат да се изпълняват с еднакъв успех и от двете страни, независимо от това, коя от тях е инициатор. Обаче в практическите решения пълната симетрия е рядко явление. По-често се среща ситуацията, когато VPN шлюзовете функционално са различни: мощен високопроизводителен сървър в централен офис очаква съединения от хиляди отдалечени клиенти, а всеки от клиентите инициира тези съединения според необходимостта и обслужва значително малка част от частна мрежа (отдалечен офис, банкомат и т.н.). Като следствие, условията за функциониране на сървъра и клиента също така могат да се различават съществено: сървърът да се намира на високонадеждно (Ethernet, Fiber Ethernet) съединение със статичен IP адрес, а в същото време клиентите могат да използват най-различни технологии за достъп до Интернет, в това число и неустойчиви линии за връзка, динамични адреси, NAT за излизане от мрежата на доставчика на услуги в Интернет, и т.н.

12. Увод в маршрутизацията

Обединената мрежа (internetwork или internet) – това е съвкупност от няколко мрежи, наричани също така подмрежи (subnet), които се свързват помежду си с маршрутизатори. Организацията на съвместно транспортно обслужване в съставната мрежа се нарича междумрежово взаимодействие (internetworking).

Една от основните функции на мрежовия слой е предаването на пакети между крайните възли в обединените мрежи (отделните подмрежи). Този слой на практика осъществява междумрежово свързване.

Маршрут се нарича последователността от маршрутизатори, през които трябва да премине пакета от източника (изпращача) до целта (получателя). Задачата за избор на маршрут от няколко възможни се решава от маршрутизаторите и крайните възли на основание на маршрутните таблици. Записите в таблиците могат да се попълват ръчно или от протоколите за маршрутизация.

Протоколите за маршрутизация (например RIP или OSPF) следва да бъдат отличавани от мрежовите протоколи (например IP или IPX). Докато първите събират и предават по мрежата само служебна информация за възможните маршрути, вторите са предназначени за предаване на потребителски данни.

Мрежовите протоколи и протоколите за маршрутизация се реализират във вид на програмни модули на крайните възли (хостовете) и на междинните – маршрутизаторите.

Маршрутизаторът представлява сложно многофункционално устройство (специализиран компютър), в задачите на което влиза: построяване на маршрутните таблици, определяне на тяхна основа на маршрута, буферизация, фрагментация и филтрация на постъпващите пакети, поддръжка на мрежовите интерфейси. Функциите на маршрутизаторите могат да се изпълняват както от специализирани устройства (като Cisco Router), така и от универсални компютри със съответстващо програмно осигуряване (снабдени с няколко мрежови карти под управлението на универсална операционна система, като например FreeBSD или GNU/Linux).

Типичният маршрутизатор работи под управлението на специализирана операционна система, оптимизирана за изпълнение на операциите за построяване на маршрутни таблици и придвижване на пакетите, според информацията в тях.

Маршрутизаторът обикновено се изгражда като мултипроцесорна схема, използва се симетрично мултипроцесиране (SMP), асиметрично мултипроцесиране (AMP) и тяхното съчетание. Най-рутинните операции за обработване на пакетите се изпълняват програмно от специализиран процесор или апаратно от големи интегрални схеми (ASIC). Действията от по-високо ниво се изпълняват от програмно универсални процесори.

Главните функции на маршрутизаторите са: избор на най-добрия път за пакетите до адреса на получателя и комутация на приетия пакет от входящия интерфейс към съответстващия изходящ интерфейс.

За алгоритмите за маршрутизация са характерни едностъпкови и многостъпкови подходи. Едностъпковите алгоритми се делят на алгоритми с фиксирана, проста и адаптивна маршрутизация. Адаптивните протоколи за маршрутизация са най-разпространени и на свой ред могат да се базират на дистанционно-векторни алгоритми и алгоритми за състоянието на връзките.

Слоят за междумрежово взаимодействие реализира концепцията комутация на пакети в режим без установяване на съединения. Основните протоколи за този слой са дейтаграмният протокол IP и протоколите за маршрутизация (RIP, OSPF, BGP и др.). Спомагателна роля изпълняват: протоколът за междумрежови управляващи съобщения ICMP, протоколът за групово управление IGMP и протоколът за разрешаване на адресите ARP.

Маршрутизацията е процес за определяне на най-добрият път (маршрут), по който пакетът може да бъде доставен до получателя.

Възможните пътища за предаване на пакетите се наричат маршрути. Най-добрите маршрути до известните получатели се съхраняват в маршрутната таблица.

В зависимост от начина на попълване на маршрутните таблици се различават два вида маршрутизация:

- Статична маршрутизация - данните се въвеждат от мрежовия администратор;

- Динамична маршрутизация - информацията постъпва от съседните маршрутизатори като се използва протокол за динамична маршрутизация.

Маршрутизаторът оценява достъпните пътища до адреса на получателя и избира най-рационалния маршрут на база на някакъв критерий - метрика. Най-малката метрика означава най-добър маршрут. Метриката на статичен маршрут винаги е равна на 0.

Процесът на маршрутизация на дейтаграмите се състои в определянето на следващия възел (next hop) по пътя на предаването на дейтаграмата и прехвърлянето й към този възел, който се определя като цел или междинен маршрутизатор.

Нито възелът изпращач, нито някой междинен маршрутизатор разполагат с информация за цялата верига, по която се предава дейтаграмата. Всеки маршрутизатор, а също така и възелът изпращач, базирайки се на адреса на получателя на дейтаграмата, определят единствено следващия възел от нейния маршрут.

Маршрутизацията на дейтаграмите се осъществява на нивото на IP протокол.

Маршрутизацията се реализира като се използват данните, съдържащи се в маршрутната таблица. Колонките в маршрутната таблица се състоят от следните полета:

- адрес на мрежа цел (мрежа на получателя);
- адрес на следващия маршрутизатор (т. е. възела, който знае накъде по-нататък да предаде дейтаграмата, която е адресирана за мрежата на получателя);
- спомогателни полета.

Видът на маршрутната таблица зависи от конкретната реализация на маршрутизатора, но въпреки големите външни разлики, във всички типове таблици на маршрутизаторите се използват ключовите полета, необходими за реализация на маршрутизацията.

Вече беше споменато, че IP модулът динамично се маршрутизира. При маршрутизацията обикновено се използва само част от информацията на IP пакета – адрес на получателя. Маршрутната таблица се използва, само когато се определя как да се доставят пакетите. Алгоритъмът на маршрутизацията се нарича най-доброто (най-дългото) съвпадение (longest prefix match). Как функционира той? Като се започне с най-дългата маска (255.255.255.255), се изпълнява операцията „логическо И“ с IPdst, докато се получи съвпадение с NETdst (мрежата на получателя). Това означава, че е открит ред от маршрутната таблица, който ще се използва за маршрутизацията на пакета.

Съществуват няколко източника, доставящи записи в маршрутната таблица. Първо, при инициализация на програмното осигуряване на TCP/IP стека се внасят в таблицата записи за непосредствено свързаните мрежи, а също така и записи за особени адреси от типа на 127.0.0.0. Второ, администраторът ръчно записва статичните записи за специфични маршрути или за адрес на шлюза по подразбиране (default gateway). Трето, протоколите за маршрутизация автоматично внасят в таблицата динамичните записи за наличните маршрути.

За да работят в мрежата крайните възли, на тях е необходимо да се конфигурира: IP адрес на възела, мрежова маска (префикс), адрес на шлюза по подразбиране. За

крайния възел (хоста), когато адресът на получателя се намира в друга мрежа, тогава той препраща пакета на шлюза по подразбиране (маршрутизатора по подразбиране). Тази роля изпълнява интерфейс на маршрутизатора, през който всички пакети от локалната мрежа се препращат в отдалечените мрежи.

Препоръчва се да се задават статично адреси на интерфейсите на маршрутизаторите, сървърите, мрежовите принтери.

Според областите на приложение маршрутизаторите се делят на: магистрални маршрутизатори, маршрутизатори на регионалните подразделения, маршрутизатори на отдалечените офиси и маршрутизатори за локални мрежи (и комутатори от 3-ти слой).

Основните характеристики на маршрутизаторите са: обща производителност (измервана в пакети за секунда), набор от поддържани мрежови протоколи и протоколи за маршрутизация, набор от поддържани мрежови интерфейси на глобални и локални мрежи.

Главната особеност на комутаторите от 3-ия слой е високата скорост на изпълнение на операциите за маршрутизация за сметка на прехвърлянето им на апаратния слой – нивото на големите интегрални схеми (ГИС/ASIC).

Към допълнителните функции на маршрутизатора се отнасят освен едновременната поддръжка на няколко мрежови протоколи и няколко протокола за маршрутизация, възможността за приоритетна обработка на трафика, разделянето на функциите за построяване на маршрутни таблици и придвижването на пакетите между маршрутизаторите от различен клас на база на използване на готовите маршрутни таблици.

Много фирми са разработили собствени протоколи за ускоряване на маршрутизацията на дълговременните потоци в локалните мрежи, които маршрутизират само няколко от първите пакети на потока, а останалите пакети се комутират на база на MAC адресите.

Корпоративните многофункционални концентратори представляват такива устройства, в които на общата вътрешна шина се обединяват модули от различен тип - повторители, мостове, комутатори и маршрутизатори. Такова обединение позволява да е възможно програмното конфигуриране на мрежата с определяне на подмрежите и сегментите без зависимост от физическото свързване към един или друг порт (интерфейс) на устройството.

За пренасочването на трафика към всички мрежи, съставящи обединената мрежа, маршрутизаторът трябва да притежава следната информация:

- IP адрес на получателя;
- IP адрес на съседния маршрутизатор, от когото той може да научи за отдалечените мрежи;
- Достъпни пътища до всички отдалечени мрежи;
- Най-добрият път до всяка отдалечена мрежа;
- Методи за обслужване и проверка на информацията за маршрутизацията.

Крупните мрежи се разбиват на автономни системи, в които се провежда обща политика за маршрутизация на IP пакетите. Ако такава мрежа е свързана с Интернет, тогава идентификаторът на автономната система се задава от InterNIC.

Протоколите за маршрутизация се делят на външни и вътрешни. Външните протоколи (EGP, BGP) пренасят маршрутната информация между автономните системи, а вътрешните (RIP, OSPF) се прилагат единствено в границите (в рамките) на определена автономна система.

Административно разстояние (AD, administrative distance) — параметър на маршрута, определящ степента на достоверност (доверие) към информацията от източника за маршрутизация. AD представлява цяло число в диапазона от 0 до 255, където 0 означава най-голямо доверие, а 255 — забрана за предаване на трафика по дадения път. Колкото е по-малко AD, толкова е по-висока достоверността. За статичните маршрути AD = 1.

Различават следните класове протоколи за динамична маршрутизация:

Протоколи с векторно разстояние (Distance vector) **—** протоколи за маршрутизация на база векторно разстояние, които използват за търсене на най-добрия път разстоянието до отдалечената мрежа. Всяко пренасочване на пакета от маршрутизатора се нарича hop (хоп, скок). За най-добър се счита пътят до отдалечената мрежа с най-малък брой хопове. Векторът определя направлението към отдалечената мрежа. Примери за протоколи за маршрутизациа с векторно разстояние са RIP и IGRP.

Протоколи със състояние на връзките (Link state) **—** обикновено се наричат "първият – най-краткия път" (SPF). Всеки маршрутизатор създава три различни таблици. Първата от тях проследява непосредственото свързване на съседите, втората — определя топологията на цялата обединена мрежа, а третата е маршрутната таблица. Устройството, действащо според протокол от типа състояние на връзките, има повече сведения за обединената мрежа, от колкото всеки протокол с векторно разстояние. Примери за протоколи за IP маршрутизация за състоянието на връзките са протоколите OSPF и IS-IS.

Не съществува единен метод за конфигуриране на маршрутните протоколи във всяка една произволна мрежова среда. Тази задача се решава като се отчитат особеностите на конкретната мрежа. Обаче знанията за разликите в работата на различните протоколи за маршрутизация може да подпомогне при избора на най-доброто решение.

Междумрежово взаимодействие за IPv6

На IPv6 интерфейсите се конфигурират глобални и локални индивидуални адреси на канала, използвани за обмен на данните вътре в локалния канал (подмрежата), където те трябва да са уникални.

Конфигурираните локални индивидуални адреси на канала позволяват на мрежовите устройства да си обменят маршрутна информация и да препращат съобщения вътре в локалния канал без прилагането на глобални адреси.

Версията IPv6 позволява да се използва в качеството на идентификатор на интерфейса неговият MAC адрес, който се конфигурира автоматично.

Маршрутизацията за IPv6 се включва след формирането на командата ipv6 unicast-routing в режим на глобално конфигуриране.

При предаването на данните през съставната мрежа IP адреса на възела получател и на възела източник остават неизменни (непроменени).

При предаването на данни през съставната мрежа MAC адресите на получателя и източника се променят при преминаването през всеки маршрутизатор.

При формирането на кадъра се изчислява контролната сума, която се записва в полето FCS на трейлъра на кадъра. При приемането на кадъра на всеки входящ интерфейс отново се изчислява контролната сума, която се сравнява с приетата.

При предаването на данни през съединение "точка-точка" заглавието на кадъра може да бъде съществено опростено.

Форматът на командата за конфигуриране на стандартна статична маршрутизация е следният:

Router(config)#ip route <адрес> <маска> <next hop>

Форматът на командата за конфигуриране на статична маршрутизация с използването на изходящ интерфейс (такъв маршрут се нарича окончателен) е следният:

Router(config)#ip route <адрес> <маска> <изходящ интерфейс>

Статичната маршрутизация по премълчаване се използва за изпращане на пакети, когато мрежата на получателя отсътства в маршрутната таблица.

Форматът на командата за конфигуриране на статична маршрутизация по премълчаване е следният:

Router(config)#ip route 0.0.0.0 0.0.0.0 <next hop>

В маршрутната таблица, създадените от администратора статични маршрути към отдалечени мрежи, са отбелязани със символа S, а маршрутите по премълчаване – със символа S*.

Верификацията на маршрутните таблици се реализира с използване на командите show ip route, show ipv6 route.

Динамична маршрутизация

Протоколите за динамична маршрутизация разделят мрежовата информация чрез обмен на данни между маршрутизаторите.

Протоколите за маршрутизация откриват мрежите и изграждат маршрутите до отдалечените мрежи, като създават и поддържат маршрутни таблици, на база на които маршрутизаторът предава постъпилия пакет от входящия интерфейс към изходящ.

В качеството на метрика може да се използва: броя на преходите между маршрутизаторите по пътя до мрежата на получателя (брой междинни рутери); честотна лента на съединенията; забавяния по пътя от източника до получателя; натоварване на канала; надеждност на предаването; обобщена стойност.

При изменение на топологиите е необходимо някакво време (време за сходимост или конвергенция) за съгласуване на информацията в маршрутните таблици на всички маршрутизатори в мрежата. Времето за сходимост трябва да бъде минимално.

Съвкупността от мрежи, представени от набор маршрутизатори под общо административно управление, образува автономна система.

13. RIP протокол

RIP (Routing Information Protocol) е лесен за настройване и управление, понеже използва дистанционно-векторния алгоритъм. RIP използва алгоритъма на Белман-Форд (Bellman–Ford). Той работи успешно в неголеми мрежи с до 15 междинни маршрутизатори.

RIP маршрутизаторите при избор на маршрута обикновено използват най-простата метрика – броят на междинните маршрутизатори между мрежите до получателя, т. е. хопове.

Версията RIPv1 не поддържа маски на подмрежи, което принуждава администраторите да използват маски с фиксирана дължина в цялата съставна мрежа. Във версия RIPv2 това ограничение е свалено.

В мрежите, използващи RIP и притежаващи маршрути с примки (routing loops), могат да се наблюдават достатъчно дълги периоди с нестабилна работа, когато пакетите се зациклят в маршрутните примки и не достигат до получателя си. За борба с тези явления в RIP маршрутизаторите са предвидени допълнителни методи (SplitHorizon, Hold Down, Triggered Updates), които съкращават в някои случаи периодите на нестабилност.

RIP протоколът не знае пълната топология на мрежата. Той знае само дистанцията (метриката) и вектора (next-hop адрес). Освен адреса на мрежата и маската на мрежата в обновленията е включен адресът next-hop и метриката на маршрута.

RIP прилага два основни типа съобщения - заявка (request) за първоначално включване за запитване на съседите и отговор (reply) за изпращане на обновления.

Характеристики на протокола:

- RIPv1 и RIPv2 използва UDP номер на порт 520.
- RIPng използва UDP номер на порт 521.
- За предаване на съобщения RIPv1 за адрес на получател използва броудкастен адрес 255.255.255.255, а RIPv2 - групов адрес 224.0.0.9.

Таймери на протокола

- *Update timer* (Таймер за обновления) — Той задава честотата на изпращане за обновления на протокола, след изтичане на таймера се изпраща обновлението. По подразбиране е равен на 30 секунди.
- *Invalid timer* (Таймер за невалидност) — Ако не е получено обновление за маршрута до изтичането на този таймер, маршрутът ще бъде отбелязан като недостижим (Invalid), т. е. с метрика 16. По подразбиране този таймер е равен на 180 секунди.
- *Flush timer / garbage collection timer* (Таймер за почистване) — По подразбиране таймерът е равен на 240 секунди, с 60 повече от invalid timer. Ако този таймер изтече преди да пристигнат обновленията за маршрута, тогава маршрутът ще се изтрие от маршрутната таблица. Ако маршрутът е изтрит от маршрутната таблица, тогава съответно се изтриват и останалите таймери, които му съответстват.
- *Holddown timer* (Таймер за задържане) — Този таймер се стартира след като маршрутът се отбележи като недостижим. Докато изтече таймерът, маршрутът ще се намира в паметта, за да се предотврати образуването на маршрутна примка и по този маршрут ще се предава трафика. По подразбиране той е равен на 180 секунди. Таймерът не е стандартен, добавен е в реализацията на Cisco.

Описание на работата на протокола

Съседът получава обновление, в което е посочено каква метрика за получения маршрут да използва. Когато маршрутизаторът получи RIP обновление, той добавя към метриката на маршрута единица.

Маршрутизаторът изпраща на всеки 30 секунди всички известни му маршрути на съседните маршрутизатори. Но, освен това, за да предотвратява примки и за подобряване на времето за сходимост, се използват допълнителни механизми:

- *Split horizon* (Расцепване на хоризонта) — ако маршрутът е достижим през определен интерфейс, тогава в обновлението, което се изпраща през този интерфейс, не се включва този маршрут;

- *Triggered update* (Тригерни ъпдейти) — обновленията се изпращат веднага при изменение на маршрута, вместо да се изчака изтичането на Update timer;
- *Route poisoning* (Отравяне на маршрута) — това е принудително изтриване на маршрута и привеждането му в състояние на задържане. Използва се за борба с маршрутните примки.
- *Poison reverse* (Обратно поправяне) — Маршрутът се отбелязва като недостижим, т. е. с метрика 16 и се изпраща в обновленията.

В обновленията на RIPv2 максималният брой маршрути в едно съобщение е 25.

Ако в маршрутната таблица има статичен маршрут по подразбиране, тогава той може да се анонсира с помощта на командата redistribute static.

Сумаризация на маршрутите

По подразбиране при настройването на RIP се включва автоматична сумаризация. При правилно планирана адресация на мрежата тя може да бъде удобна. Има смисъл тя да се изключва, понеже RIP се реализира единствено по класова граница, т.е. сумаризация на маршрути с маска по-малка от класовата не е допустима. Също така RIP позволява да се обявяват сумарни маршрути ръчно, но по-горе описаното ограничение остава в сила. За решаването на този проблем може да се използва сумарен статичен маршрут с редистрибуция на неговите съседи (RIP поддържа сумарни статични супернет маршрути и ги записва в маршрутната таблица). За целта се използва командата redistribute static.

Маршрутизаторът може да сумира мрежи:

- автоматично, сумирайки подмрежи в класова мрежа на границата на класова мрежа (auto-summary),
- в съответствие с настройките, анонсирайки посочената мрежа за интерфейса.

База от данни за RIP маршрутите

В базата от данни се съхраняват следните маршрути:

- всички маршрути, които са били получени от RIP протокол;
- всички непосредствено присъединени мрежи, които RIP анонсира за съседите;
- сумарните маршрути.

Ако RIP маршрутът не може да бъде записан в маршрутната таблица (съществува друг маршрут с по-добра стойност на AD), тогава той не се съхранява в базата от данни на RIP маршрутите.

Понеже по подразбиране RIP разпраща обновления през всички интерфейси, се препоръчва през тези интерфейси, където не е необходимо това да се прави, да се

забранява, дори това да нарушава безопасността на мрежата. За целта се прилага например командата passive-intreface fa 0/0. Така RIP протокол ще спре да изпраща от дадения интерфейс обновления, но ще продължи да го използва. За да се закърпи тази дупка в безопасността, може да се настрои авторизация между маршрутизаторите по md5. Т. е. обновленията, които се получават без авторизация, ще се игнорират.

За обявяване на анонсируемите мрежи се използва командата network address. Адресът се записва в класов вид, без посочване на маската. RIP протокол ще вземе маската от съответния интерфейс.

За работа в IPv6 мрежи на базата на RIPv2 е разработен протокола RIPng.

14. OSPF протокол

Протоколът OSPF (Open Shortest Path First) е бил разработен с цел ефективна маршрутизация на IP пакети в големи мрежи със сложна топология, включваща примки. Той се базира на алгоритъма за състоянието на връзките (алгоритъма на Дикстра, Dijkstra's algorithm), който притежава висока устойчивост при изменения в топологията на мрежата.

При избор на маршрута OSPF маршрутизаторите използват метрика, отчитаща пропускателната способност на обединените мрежи.

OSPF е първият маршрутизиращ протокол за IP мрежи, който отчита битовете за качество на обслужване (пропускателна способност, забавяния и надеждност) в заглавието на IP пакета. За всеки тип качество на обслужването се построява отделна маршрутна таблица.

OSPF протоколът притежава висока изчислителна сложност, затова най-често работи на мощни апаратни маршрутизатори.

OSPF се инкапсулира в IP. Номерът на протокола (в заглавието на IP) е 89.

За предаване на пакетите се използват следните групови адреси:

- 224.0.0.5 от всички OSPF маршрутизатори;
- 224.0.0.6 от всички DR и BDR.

OSPF е представител на семейството Link-State протоколи. Административното разстояние на OSPF протокола е равно на 110.

Формат на заглавието на OSPF пакета

- Version number – версия на протокола;

- Type - определя типа на OSPF пакета: 1 – Hello, 2 – DBD, 3 – LSR, 4 – LSU, 5 - LSAck;
- Packet length - дължина на OSPF пакета в байтове (включва и заглавието);
- Router ID - определя маршрутизатора, източник на пакета (IP адреса на един от интерфейсите);
- Area ID - определя зоната, откъдето е бил генериран пакета;
- Checksum – използва се за определяне на грешките в пакета, които са възниквали по време на предаването му;
- Authentication type – тип на автентификацията: 0 - без автентификация, 1 - автентификация с отворени пароли, 2 - MD5 автентификация;
- Authentication - данни за автентификацията на маршрутите;
- Data – Полето с данни е различно за различните OSPF пакети:

    - Hello packets - съдържа списък на известните съседи;
    - DBD packet - съдържа сумарната LSDB, в която са включени: всички известни Router ID, техните *последни номера на последователностите (sequence number*) и друга информация;
    - LSR packet - съдържа типа на необходимата заявка LSU и Router ID, от когото е необходимо обновление;
    - LSU packet - съдържа целия ред LSA. Няколко записа LSA могат да се запишат в един пакет с OSPF обновления.
    - LSAck packet – празно поле за данни.

Терминология на OSPF протокола

Базови термини:

- Канал/интерфейс (link/interface) — съединение на маршрутизатора и една от свързаните с него мрежи. При обсъждането на OSPF термините интерфейс и канал (link) често пъти се употребяват като синоними;
- Метрика (metric) — условен показател на разстоянието до мрежата на получателя;
- Цена (cost) — условен показател на "цената" на препращането на данните по канала. В OSPF зависи от пропускателната способност на интерфейса (bandwidth);
- Автономна система (autonomous system, AS) — група от маршрутизатори, обменящи си маршрутна информациа с помощта на някой от протоколите за маршрутизация.

Основни термини за OSPF:

- Идентификатор на маршрутизатора (router ID, RID) — 32 битово число, което уникално идентифицира маршрутизатор в границите на една автономна система;
- Зона (area) — съвкупност от мрежи и маршрутизатори, притежаващи един и същи идентификатор на зоната;
- Обявление за състоянието на канала (link-state advertisement, LSA) — единица от данни, която описва локалното състояние на маршрутизатора или мрежата. Например за LSA маршрутизатора включва описанието на състоянието на каналите и отношението на съседство. Множеството от всички LSA, описващи

маршрутизаторите и мрежите, образуват базата от данни за състоянието на каналите (LSDB);

- База от данни за състоянието на каналите (link state database, LSDB) — списък от всички записи за състоянието на каналите (LSA). Среща се и термин топологическа база от данни (topological database), който се употребява като синоним на база от данни за състоянието на каналите.

OSPF съседи:

- Съседи (neighbours) — два маршрутизатора, интерфейсите на които се намират в един броудкастен сегмент и на които е включен OSPF на тези интерфейси;
- Отношения на съседство (adjacency) — взаимосвързаност между съседни маршрутизатори, установена с цел синхронизация на информацията;
- Hello-протокол (hello protocol) — протокол, използван за установяване и поддръжка на съседски отношения;
- База от данни на съседите (neighbours database) — списък от всички съседи (също така се използва и термина neighbour table).

OSPF пакети:

- Hello — пакети, които се използват за откриване на съседи, установяване на съседски отношения и мониторинг на тяхната достъпност (keepalive);
- DBD (Database Descriptions) — пакети, които описват съдържанието на LSDB;
- LSR (Link State Request) — пакети, с помощта на които се запитва пълната информация за LSA, които недостигат в LSDB на локалния маршрутизатор;
- LSU (Link State Update) — пакети, които предават пълната информация, която се съдържа в LSA;
- LSAck (Link State Acknowledgment) — пакети, с помощта на които се потвърждава получаването на другите пакети.

Описание на работата на протокола

1. Включване на OSPF на маршрутизатора.
2. Маршрутизаторът си избира Router ID (уникално име на маршрутизатора).
3. Включване на OSPF на интерфейсите (за да знае протоколът за кои интерфейси може да съобщи на другите маршрутизатори).
4. Откриване на съседите с помощта на Hello пакети:

    1. Маршрутизаторите си обменят hello пакети през всички интерфейси, на които е активиран OSPF.
    2. Маршрутизаторите, които се намират в един броудкастен сегмент, стават съседи, когато те достигат до договаряне на определени параметри, зададени в техните hello пакети.

5. Adjacency (отношения на съседство) - тип съседство между маршрутизаторите, според който те синхронизират LSDB. Установяването на тези отношения зависи от типа на мрежата:

1. Ако маршрутизаторите се намират в мрежа с множествен достъп, те избират DR и изпълняват LSDB синхронизация с него.

2. Ако маршрутизаторите се намират в point-to-point мрежа, те пристъпват към LSDB синхронизация по между си.

6. LSDB синхронизация. Преминава през няколко етапа. Според формираните отношения на съседство се реализира обмен на следните пакети:

1. DBD (кратко описание на LSA в LSDB). С помощта на тези пакети маршрутизаторите си съобщават взаимно, с каква информация разполагат в съкратен вид;
2. LSR. След обмяната на DBD пакети, с помощта на LSR маршрутизатори съседът запитва (от съседите си) за недостигащата информация;
3. LSU (съдържа пълно описание на LSA). Като отговор на LSR, който му е изпратил съседа, маршрутизаторът изпраща LSU, с пълно описание на информацията, която няма съседа;
4. LSAck. След получаването на LSU от съседа, маршрутизаторът изпраща потвърждение, че той е получил информацията;
5. Ако двата маршрутизатора трябва взаимно да запитват един от друг информация, тогава тази процедура се повтаря и в другата посока.
6. След това LSDB е синхронизирана, а това означава, че е напълно идентична между съседите.

7. След синхронизацията на LSDB, маршрутизаторът изпраща обновления по-нататък на своите съседи в другите броудкастни сегменти.

8. Разпращайки обявления през зоната, всички маршрутизатори си построяват идентична LSDB.

9. Когато базата от данни е построена, всеки маршрутизатор използва SPF (shortest path first) алгоритъма за изчисляване на граф без цикли (примки), който ще описва най-краткия път до всеки един известен пункт за дестинация от себе си в качеството на корен. Този граф представлява дървото на най-краткия път.

10. Всеки маршрутизатор построява маршрутна таблица, базирана на своето дърво на най-краткия път.

OSPF протоколът формира три бази от данни: база от данни на свързванията; база от данни за състоянието на каналите (LSDB); база за прехвърлянията.

На основата на тези бази от данни се формират: таблицата на съседните устройства; таблица на топологията на мрежата и маршрутна таблица.

Избор на Router ID

При стартиране на OSPF процеса на всеки маршрутизатор, задължително трябва да е избран Router ID.

Router ID е уникално име на маршрутизатора, по което той е известен в AS.

В зависимост от реализацията Router ID може да се избира различно:

- минимален IP адрес или максимален IP адрес, който е зададен на интерфейсите на маршрутизатора;
- обикновено има начин да се зададе Router ID ръчно;
- най-важното е Router ID да е уникален в AS.

След промяна на Router ID, OSPF процесът трябва да се рестартира, а всички LSA, които е генерирал този маршрутизатор, трябва да се изтрият от AS преди презареждането.

Съседи и установяване на отношения на съседство

Откриването на съседите започва след като:

1. Протоколът е бил включен глобално;
2. Избран е Router ID;
3. OSPF е включен на интерфейсите.

За откриване и мониторинг на съседи се използват Hello съобщения.

Процедурата за установяване на отношения на съседство зависи от типа на мрежата, в която работи OSPF.

Типове мрежи, поддържани от OSPF протокол

- Броудкастни мрежи с множествен достъп: Ethernet;
- Точка-точка (point-to-point): Тунели, T1, E1, PPP, HDLC, Frame-Relay P-to-P;
- Неброудкастни мрежи с множествен достъп (Non Broadcast Multiple Access, NBMA): Frame-Relay, ATM, X.25.

В различните типове мрежи работата на OSPF се различава. В това число се отличава и процесът на установяване на отношения на съседство и настройката на протокола.

На практика най-често се използват два типа мрежи:

- point-to-point;
- broadcast.

За broadcast и nonbroadcast мрежи (т. е. за мрежи с множествен достъп) се избират DR и BDR.

Като правило типът на мрежата се определя автоматично според типа на интерфейса. Но може да бъде зададен и ръчно.

Отношения на съседство (adjacency)

Различаваме понятията съсед и отношения на съседство:

- Съсед (neighbor) — два маршрутизатора, които се намират в един броудкастен сегмент и за които са съвпаднали необходимите полета в hello пакетите;
- Отношения на съседство (adjacency или full adjacency) — два съседа, които са приключили с процеса на LSDB синхронизация помежду си.

За да могат маршрутизаторите да станат съседи е необходимо:

- в hello пакетите да съвпаднат стойностите на следните полета:
    - Hello Interval — честота на изпращане на Hello съобщенията;
    - Router Dead Interval — период от време, след изтичането на който, съседът се счита за недостъпен щом не са получени Hello пакети;
    - Area ID — понеже в OSPF границата на зоната преминава през маршрутизатор, затова маршрутизаторите в един броудкастен сегмент трябва да бъдат в една зона;
    - Authentication — парола, използвана за автентификация и тип на автентификацията. Маршрутизаторите не са длъжни да използват автентификация, но в случай, че използват, паролите са според типа на автентификацията;
    - Stub area flag — незадължителен флаг, който се установява на всички маршрутизатори, които принадлежат на stub зоната.
- за маршрутизаторите трябва да съвпадат мрежата и маската на мрежата.

OSPF не проверява мрежата и маската на мрежата при установяване на отношения на съседство в point-to-point мрежи. Затова могат да се използват IP unnumbered интерфейсите.

За да могат маршрутизаторите да установят отношения на съседство, трябва да съвпадат стойностите на MTU на интерфейсите. Информацията за стойността на MTU се предава в DD (database description) пакетите и се сравнява в началото на обмена с DD пакети.

Отношенията на съседство се установяват само за primary адреси.

На интерфейса може да бъде настроен и secondary адрес. Маршрутизаторите не изпращат hello пакети от secondary адрес, не установяват отношения на съседство за secondary адреси, но мрежата на secondary адреса може да се анонсира.

Възможни състояния

1. Down — начално състояние на процеса за откриване на съседи. Това състояние определя факта, че от съседите не е била получена свежа информация.

2. Attempt — това състояние има смисъл само за съседи, които са присъединени към NBMA мрежи. То говори за това, че от съседа не е била получена свежа информация и че е необходимо да се направи опит за свързване със съседа. Последното се реализира с изпращане на съседа на Hello съобщения през Hello Interval времеви интервал.
3. Init — състояние, в което се намира маршрутизаторът, изпратил на съседа си hello и очакващ от него ответен hello.
4. Two-way — при получаването на hello пакети маршрутизаторът трябва да открие в тях своя RID в списъка на съседите. Ако това е налице, тогава той установява отношения и преминава в състояние two-way.
5. Exstart — маршрутизаторите определят Master/Slave отношения на база на Router ID. Маршрутизаторът с по-голям RID става Master маршрутизатор, който определя DD Sequence number, а също така и пръв започва обмена на DD пакети.
6. Exchange — маршрутизаторите изпращат един на друг DD пакети с информация за мрежите, съдържащи се в тяхната собствена LSDB.
7. Loading — Ако маршрутизаторът вижда, че части от маршрутите липсват в неговата LSDB, той изпраща LSR съобщение с изброяване на тези мрежи, за които той иска да получи допълнителна информация. Докато маршрутизаторът се намира в очакване на отговора във вид на LSU съобщения, той пребивава в състояние Loading.
8. Full — Когато маршрутизаторът е получил цялата информация и LSDB на двата маршрутизатора е синхронизирана, двата маршрутизатора преминават в състояние fully adjacent (FULL).

Отговорен маршрутизатор (DR) и резервен отговорен маршрутизатор (BDR)

В мрежите с множествен достъп отношенията на съседство трябва да бъдат установени между всички маршрутизатори. Последното води до това, че се разпращат голям брой копия на LSA. Например, ако броят на маршрутизаторите в мрежа с множествен достъп е n, тогава ще бъдат установени $n(n-1)/2$ отношения на съседство. Всеки маршрутизатор ще разпрати $n-1$ LSA на своите съседи, плюс един LSA за мрежата, като резултат мрежата ще генерира n^2 LSA.

За предотвратяването на проблема с разпращане на LSA копия в мрежите с множествен достъп се избират DR и BDR.

Отговорен маршрутизатор (designated router, DR) — управлява процеса на разпращане на LSA в мрежата. Всеки маршрутизатор в мрежата установява отношения на съседство с DR. Информацията за измененията в мрежата се изпраща на DR от маршрутизатора, открил това изменение, а DR отговаря тази информация да бъде изпратена на останалите маршрутизатори в мрежата.

Недостатъкът в схемата на работа с DR маршрутизатор е това, че при излизането му от строя, трябва да бъде избран нов DR. Новите отношения на съседство трябва да бъдат формирани и, докато базите от данни на маршрутизаторите не се синхронизират с базата от данни на новия DR, мрежата ще бъде недостъпна за препращане на пакети. За отстраняването на този недостатък се избира BDR.

Резервен отговорен маршрутизатор (backup designated router, BDR). Всеки маршрутизатор в мрежата установява отношения на съседство не само с DR, но и с BDR. DR и BDR също така установяват отношения на съседство и помежду си. При излизане от строя на DR, BDR става DR и изпълнява всички негови функции. Понеже маршрутизаторите в мрежата са установили отношения на съседство с BDR, затова времето на недостъпност на мрежата се минимизира.

Маршрутизаторът, избран за DR или BDR в една от присъединените му мрежи с множествен достъп, може да не бъде DR (BDR) в друга присъединена мрежа. Ролята на DR (BDR) представлява свойство на интерфейса, а не свойство на маршрутизатора.

След като бъдат избрани, DR и BDR запазват своите роли, даже в случай, че към мрежата се добавят други маршрутизатори с по-висок приоритет до тогава, докато маршрутизаторите не се преконфигурират.

Таймери на протокола

- HelloInterval — Интервал от време в секунди, след изтичането на който маршрутизаторът изпраща следващ hello пакет от интерфейса. За броудкастни мрежи и за мрежи от типа точка-точка стойността по подразбиране и обикновено е равна на 10 секунди. За неброудкастни мрежи с множествен достъп стойността по подразбиране е 30 секунди.
- RouterDeadInterval — Интервал от време в секунди, след изтичането на който съседът ще се счита за недостижим (dead). Този интервал трябва да бъде кратен на стойността на HelloInterval. Като правило RouterDeadInterval е равен на 4 интервала за изпращане на hello пакети, т. е. 40 секунди.
- Wait Timer — Интервал от време в секунди, след изтичането на който маршрутизаторът си избира DR в мрежата. Тази стойност е равна на стойността на интервала RouterDeadInterval.
- RxmtInterval — Интервал от време в секунди, след изтичането на който маршрутизаторът повторно ще изпрати пакет, на който не е получил потвърждение за получаването (например Database Description пакет или Link State Request пакети). Този интервал се нарича още Retransmit interval. Стойността на интервала е 5 секунди.

Типове маршрутизатори

- *Вътрешен маршрутизатор (internal router)* — маршрутизатор, всички интерфейси на който принадлежат на една зона. Такъв маршрутизатор има само една база от данни за състоянието на каналите.
- *Граничен маршрутизатор (area border router, ABR)* — съединява една или повече зони с магистралната зона и изпълнява функции на шлюз за междузоналния трафик. Граничният маршрутизатор винаги има поне един интерфейс,

принадлежащ на магистралната зона. За всяка присъединена зона маршрутизаторът поддържа отделна база от данни за състоянието на каналите.

- *Магистрален маршрутизатор (backbone router)* — маршрутизатор, на който винаги поне един от интерфейсите принадлежащи на магистралната зона. Така определението прилича на това, за граничния маршрутизатор, обаче, магистралният маршрутизатор не винаги е граничен. Вътрешният маршрутизатор, интерфейсите на който принадлежат на нулевата зона, също така е магистрален.
- *Граничен маршрутизатор на автономната система (AS boundary router, ASBR)* — той си обменя информация с маршрутизатори, принадлежащи на други автономни системи или не-OSPF маршрутизатори. Граничният маршрутизатор за автономната система може да се намира на всяко едно място в автономната система и да бъде вътрешен, граничен или магистрален маршрутизатор.

OSPF зони

При разделянето на автономната система на зони, на маршрутизаторите, принадлежащи на една зона, е неизвестна информацията за детайлната топология на другите зони.

Разделянето на зони позволява:

- Да се намали натоварването (на цифрово-програмното управление) на маршрутизаторите за сметка на намаляването на броя на преизчисленията по SPF алгоритъма;
- Да се намали размера на маршрутните таблици (за сметка на сумирането на маршрутите на границите на зоните);
- Да се намали броя на пакетите за обновления за състоянието на канала.

На всяка зона се присвоява идентификатор на зоната (area ID). Идентификаторът може да бъде зададен в десетичен формат или във формат на запис на IP адрес. Обаче идентификаторите на зоните не се явяват IP адреси и могат да съвпадат с всеки зададен IP адрес.

В OSPF взаимодействието между зоните е възможно само през зона 0:

- в зона 0 не бива да има разкъсвания;
- ако ненулева зона трябва да се присъедени към друга ненулева, тогава се използва virtual-link или обикновен тунел настроен ръчно.

Магистрална зона (backbone area)

Магистралната зона (известна още като нулева зона или зона 0.0.0.0) формира ядрото на OSPF мрежата. Всички останали зони са свързани с нея и междузоналната маршрутизация се реализира чрез маршрутизатор, съединен с магистралната зона.

Магистралната зона е отговорна за разпространението на маршрутната информация между немагистралните зони. Магистралната зона трябва да бъде съседна (непосредствено свързана) с другите зони, но не е задължително тя да бъде физически съседна. Съединението с магистралната зона може бъде установено и с помощта на виртуални канали.

Стандартна зона (standard area)

Обикновено зона, която се създава по подразбиране. Тази зона приема обновления за каналите, сумарните маршрути и външните маршрути.

Stub area

- Тя не приема информация за външни маршрути за автономната система, но приема маршрути от други зони.
- Ако на маршрутизаторите от stub зоната е необходимо да предават информация навън от границите на автономната система, тогава те използват маршрут по подразбиране.
- В stub зоната не може да се намира ASBR.
    - Има изключение от това правило: ABR може да бъде и ASBR.
- На всички маршрутизатори в тази зона трябва да е зададено stub.

Totally stubby area

- Тя не приема информация за външни маршрути на автономната система и за маршрути от другите зони.
- Ако на маршрутизаторите от stub зоната е необходимо да се предава информация извън границата на зоната, тогава те използват маршрут по подразбиране.
- В totally stub зоната не може да се намира ASBR.
    - Изключение от това правило: ABR може да бъде и ASBR.
- На всички маршрутизатори в зоната трябва да е зададено stub.
    - Замяната на междузоналните маршрути на маршрут по подразбиране (totally stubby) се настройва само на ABR зоните.

Фактически totally stub зоната, това е "усилване" на stub: в нея не само външните маршрути, но и междузоналните са заменени на маршрут по подразбиране.

Not-so-stubby area (NSSA)

- Тя работи на същите принципи, както и stub зоната:
    - Единствената разлика е в това, че в NSSA зоната може да има ASBR.
    - Външните маршрути от другите зони също така са заменени с маршрут по подразбиране.

- Понеже в RFC е определено, че в stub зоната не може да има ASBR, следователно LSA 5 за NSSA зоната е създаден специален тип LSA: LSA type 7.
- LSA 7 предава външни маршрути в NSSA зоната и по всичко съответства на LSA 5.

    - Когато граничен маршрутизатор на NSSA зона предава LSA 7 в други зони, тогава вместо LSA 7 се предава стандартен LSA 5.

Totally NSSA

- Тя работи на същите принципи, както и NSSA:

    - Единствената разлика е в това, че в totally NSSA зоната всички маршрути на другите зони и външните маршрути за AS, се заменят на маршрут по подразбиране.

Обявления за състоянието на канала (Link State Advertisement, LSA)

Обявлението за състоянието на канала — това е единица от данни, която описва локалното състояние на маршрутизатора или мрежата.

Множеството от всички LSA, описващи маршрутизаторите и мрежите, образуват база от данни за състоянията на каналите (LSDB).

Всеки тип LSA си има своя собствена функция:

- Router LSA и Network LSA описват как са съединени маршрутизаторите и мрежите вътре в зоната.
- Summary LSA са предназначени за съкращаване на количеството на предаваната информация за зоните. Описват мрежите от другите зони за локалната.
- ASBR Summary LSA описва за другите зони как да достигнат до локалния ASBR.
- AS External LSA позволява да се предава в автономната система информация, която е получена от външни източници (например от друг протокол за маршрутизация).

Фактически, маршрутизаторите не предават LSA. Маршрутизаторите предават LSA вътре в други пакети:

- В Database Description се предава описание на всички LSA, които се съхраняват в LSDB маршрутизатора;
- В Link State Request се предава запитване с описание на тези LSA, които не достигат в LSDB;
- В Link State Update се предават пълните LSA;
- В Link State Acknowledgment се предават потвърждения за получаването на конкретни LSA, с описанието им.
-

Табл. 2 **Обща информация за LSA**

№ на LSA	**Наименование на LSA**	**Link-State ID**	**Изпращат се от**	**Област на разпространение**
LSA 1	Router LSA	Router ID на изпращача	Всички маршрутизатори	Вътре в зоната (IntraArea)
LSA 2	Network LSA	IP адрес на интерфейса, който е DR	DR в мрежи с множествен достъп	Вътре в зоната (IntraArea)
LSA 3	Network Summary LSA	Мрежа цел и маска на мрежата	ABR	AS (InterArea)
LSA 4	ASBR Summary LSA	Router ID ASBR	ABR	AS (InterArea)
LSA 5	AS External LSA	Външна мрежа и маска	ASBR	AS (InterArea)
LSA 7	AS External LSA for NSSA	Външна мрежа и маска	ASBR само в NSSA	NSSA
LSA 6, 8, 9, 10, 11	Opaque LSAs			

Типове OSPF пакети

OSPF използва 5 типа пакета:

- Hello — използва се за откриване на съсед, за построяване на отношения на съседство с него и за мониторинг на достъпността му.
- Database Description (DBD) — проверява синхронизацията на базата от данни между маршрутизаторите.
- Link-State Request (LSR) — запитва определени записи за състоянието на каналите от маршрутизатор към маршрутизатор.
- Link-State Update (LSU) — изпраща определени записи за състоянието на каналите като отговор на запитването.
- Link-State Acknowledgment (LSAck) — потвърждава получаването на други типове пакети.

Инкрементиране на SPF и частични изчисления

При получаването на Network Summary LSA маршрутизаторът добавя в маршрутната таблица информация за мрежите, които се анонсират с тези LSA, но не стартира SPF алгоритъма за тези мрежи.

Метриката за тези мрежи се изчислява като се вземе стойността, която се анонсира в Network Summary LSA, плюс цената на пътя до ABR, който е изпратил LSA.

Ако в зоната са настъпили изменения, тогава маршрутизаторите в другите зони не стартират SPF, а използват нова метрика, която стои в Network Summary LSA, като добавят към нея цената на пътя до ABR и записват този маршрут в маршрутната таблица. Това се нарича Partial SPF calculation.

Partial SPF calculation се изпълнява независимо от това дали е настроено сумиране на маршрутите на границата на зоната или не е.

Сходимостта или конвергенцията (convergence) на мрежата е достигната, когато базите от данни за състоянието на каналите в LSDB са еднакви за всички маршрутизатори в зоната.

Избор на най-добрия маршрут

Маршрутизаторът избира най-добрия маршрут на база на най-малката стойност на метриката. Обаче OSPF отчита и някои други фактори при избора на маршрут.

Избор на най-добрия тип маршрут

Ако на маршрутизатора са известни маршрути до една и съща мрежа, но тези маршрути са от различен тип, тогава маршрутизаторът избира най-приоритетния тип маршрут и не отчита цената на маршрута.

Различните типове маршрути, подредени според намаляването на големината на приоритета са:

- Вътрешни маршрути за зоната (intra-area);
- Маршрути между зоните (interarea);
- Външни маршрути от тип 1 (E1);
- Външни маршрути от тип 2 (E2).

Въпреки че цената на маршрута E2 не се променя при предаването му по зоните (не се прибавя цената на пътя към ASBR), при съвпадането на цените на маршрутите E2, се сравнява цената на пътя до този ASBR, който анонсира маршрута.

Метрики на OSPF и тяхната цена

OSPF използва метрика, която се нарича цена (cost). Цената се сравнява за маршрутите от един тип.

В RFC 2328 не се описва как да се определя цената на интерфейса. Определен е само диапазонът от стойности: от 1 до 65535.

В OSPF метриката представлява оценка на ефективността на връзката в свързващата линия: колкото е по-малка стойността на метриката, толкова е по-ефективна организацията на връзката. Стойността на пълния (съставния) маршрут е равна на сумата от стойностите на метриките за всички свързвания, влизащи в маршрута. Сумарната цена на маршрута се изчислява като се сумират цените на изходящите интерфейси по пътя на предаването на LSA.

За да се обозначи недостижима мрежа, OSPF използва метрика равна на 16777215 ($= 2^{24} —1$).

За всяка от метриките OSPF протоколът построява отделна маршрутна таблица. Най-често в OSPF се избира маршрут на база пропускателна способност на канала. Част от метриките имат стандартизирана последователност за изчисляването на стойността, но за метриките, оценяващи надеждността, забавянията и цената на предаването, засега тази последователност не е определена. Тези въпроси се решават от администратора на мрежата.

1. Метрика – пропускателна способност на канала.

Reference bandwidth (еталонна) може да се определя от производителите на маршрутизаторите по различен начин.

Например в Cisco цената на интерфейса се изчислява по формулата:

cost = reference bandwidth / link bandwidth

Reference bandwidth е пропускателната способност, относително която се изчислява цената на интерфейса по подразбиране. Тя обикновено е 100Mb, но може да се променя.

CISCO определя тази метрика като брой секунди, необходими за предаването на 100Mb информация по канала на мрежата.

Например според тази формула: за канал със скорост 100Mb/s съответства метрика 1; за мрежа Ethernet / 802.3 (10Mb/s) съответства метрика 10.

2. Метрика - хопове. В най-простия случай стойността на метриката на маршрута може да се равнява на неговата дължина в препращания (hops), както е за RIP протокол.

3. Метрика – натоварване на канала. Натоварването на канала се променя в зависимост от интензивността на използването на канала и затова за маршрутизацията понякога е целесъобразно избирането на най-малко натоварените канали. В случая, когато има няколко маршрута с еднакви стойности на метриката, маршрутизаторите могат да използват за предаването на пакетите всички тези маршрути, осигурявайки балансирано натоварване. OSPF маршрутизаторът записва в маршрутната таблица всички маршрути с еднакви (или близки) метрики и така балансирането на натоварването между маршрутите става автоматично.

4. Метрика – забавяне. Забавянето се определя от времето за разпространение на пакета в микросекунди, което е необходимо на маршрутизатора за обработване, престой в опашките и предаването на пакетите.

5. Метрика – цена. Определя се цената на предаването на единица данни през дадения канал, възможно е например в зависимост от часа на денонощието.

6. Метрика – тип на услугата. OSPF протокол позволява да се определи за всяка една мрежа стойността на метриката в зависимост от типа на услугата ToS (Type of Service). За последните версии на OSPF ToS не се използва.

7. Други метрики. Такива са надеждност на връзката; брой дейтаграми, стоящи в опашката за предаване; изисквания за безопасност; възможности за междинни свързвания (многовариантност за достигане на получателя).

ABR Loop Prevention

Вътре в зоните OSPF използва логиката на link-state протокола, но между зоните OSPF работи като дистанционно-векторен протокол.

Например при анонсиране в зоната type 3 LSA се предава информация за мрежата на получателя, цената на маршрута и ABR, през който тази мрежа е достижима. Параметрите са аналогични на информацията, която предават дистанционно-векторните протоколи.

OSPF не използва традиционните механизми на дистанционно-векторните протоколи за предотвратяване на примки. OSPF използва няколко правила, които касаят разпространението на LSA между зоните и по този начин се изключва възможността за възникване на примки. Но последното може да доведе това, че предаването на данните ще се реализира не по най-добрия маршрут.

Външни маршрути

OSPF използва два типа маршрути за описването на мрежите извън автономната система на маршрутизатора:

- Type 1 external routes (E1);
- Type 2 external routes (E2).

Type 1 external routes — към метриката на външния маршрут се добавя цената на пътя до ASBR, който анонсира този маршрут. Използва се, когато няколко маршрутизатори анонсират външната мрежа. Когато ABR предава type 5 LSA в друга зона, той създава type 4 LSA, което определя цената на пътя от този ABR до ASBR, който е създал type 5 LSA.

Маршрутизаторът (а не ABR), който се намира в различни зони с ASBR, ще изчисли метриката на външния маршрут E1 като събере следните стойности на метриките:

- Метриката на външния маршрут, която е зададена в type 5 LSA;

- Цената на пътя до ASBR, която се анонсира в type 4 LSA;
- Цената на пътя до ABR, която е анонсирал type 4 LSA.

Type 2 external routes (използва се по подразбиране за външните маршрути) — използва се цената на външния маршрут и при предаване по мрежата цената не се увеличава. Другите маршрутизатори, при получаването на type 5 LSA, просто добавят в своята маршрутна таблица маршрута до външната мрежа с цената, която е зададена в type 5 LSA.

Макар че цената на E2 маршрута не се променя при предаването му по зоните (не се добавя цената на пътя до ASBR), при съвпадането на цените на E2 маршрутите, се сравняват цените на пътищата до ASBR, които анонсират маршрута.

Forwarding address в Type 5 LSA

Ако forwarding address не е равен на 0.0.0.0, тогава се прави проверка в маршрутната таблица. Маршрутът към този адрес трябва да бъде вътрешен за зоната или между зоните, и не може да бъде външен. Иначе първоначалният външен маршрут не се отчита.

С други думи, не може да се използва външен маршрут с цел да се достигне до друг външен маршрут. Това може да доведе до примки и затова е забранено.

Изчисляване на маршрутната таблица

Като се ползва базата от данни за състоянието на каналите за зоните, с които е свързан, маршрутизаторът изпълнява описаната по-долу последователност от действия, като така построява стъпка по стъпка маршрутната таблица. На всеки етап маршрутизаторът се обръща към определени участъци от LSDB. Ако в LSDB има LSA, за които LS age е равно на MaxAge, тогава те не се отчитат при изчисляването на маршрутната таблица.

Процес на построяване на маршрутната таблица:

1. Текущата маршрутна таблица се нулира (построяването започва от нулата). Старата маршрутна таблица се запазва (единствено), за да може да се открият измененията в определени записи на таблицата.
2. След построяването на дървото на най-краткия път, за всяка една присъединена зона се изчисляват маршрутите вътре в зоната. По време на изчисляването на дървото на най-краткия път за зоната, за зоната се изчисляват TransitCapability, което се използва по-късно на 4 етап. Фактически всички записи на маршрутната таблица с тип на целта (Destination Type) area border router се изчисляват на втория етап. Този етап се състои от две части:

    1. Първо дървото се строи като се отчитат само свързванията между маршрутизаторите и транзитните мрежи.
    2. После се включват в дървото и stub мрежите.

3. Междузоналните маршрути се изчисляват като се преглеждат съществуващите summary LSA. Ако маршрутизаторът е граничен, тогава се преглеждат сумарните LSA само на магистралната зона.
4. На граничните маршрутизатори, които са присъединени към една или повече транзитни зони (немагистрални зони, в които TransitCapability е установено в TRUE), се проверяват сумарните LSA за транзитните зони. LSA се проверяват за наличие на по-добри пътища от тези, които са били открити на етапите 2 и 3.
5. Изчисляват се маршрутите към външните мрежи. За целта се преглеждат AS-external-LSA. Местоположението на ASBR маршрутизаторите е вече определено на етапите от 2 до 4.

OSPF за IPv6 - OSPF version 3

В IPv6 мрежите при конфигуриране на OSPF3 протокол включването на маршрутизацията се осъществява с командата ipv6 unicast-routing в режим на глобално конфигуриране. На маршрутизатора се задава идентификатор, например 1.1.1.1 с командите:

R-A(config)#ipv6 router ospf 1

R-A(config-rtr)#router-id 1.1.1.1

Включването на OSPF3 протокол на интерфейсите се реализира с командата:

Router(config-if)#ipv6 ospf 1 area 0

За проверка на създадените от OSPF3 протокол маршрути се прилага командата:

show ipv6 route

15. BGP протокол

BGP (Border Gateway Protocol) е основният протокол за външна динамична маршрутизация, който се използва в Интернет.

Маршрутизаторите, използващи BGP протокол, си обменят информация за достижимостта на мрежите. Заедно с информацията за мрежите се предават различни атрибути на тези мрежи, с помощта на които BGP избира най-добрия маршрут и настройва политиките за маршрутизация.

Един от основните атрибути, който се предава с информацията за маршрута, е списъкът на автономните системи, през които е преминала тази информация. Последното позволява на BGP да определя къде се намира мрежата относно автономните системи, да

се изключват примки в маршрутизацията, а също така да може да бъде използван при настройването на политиките.

Маршрутизацията се осъществява постъпково от една автономна система към друга. Всички BGP се настройват основно по отношение на външните/съседните автономни системи, т. е. се описват правила за взаимодействие с тях.

Понеже BGP оперира с големи обеми от данни (текущият размер на таблиците за IPv4 е повече от 500 000 маршрута), затова принципите на неговата настройка и работа се отличават от тези, за вътрешните протоколи за динамична маршрутизация.

Терминология на протокола

- *Вътрешен протокол за маршрутизация (interior gateway protocol, IGP)* – протокол, който се използва за предаване на информация за маршрутите вътре в автономната система.
- *Външен протокол за маршрутизация (exterior gateway protocol, EGP)* – протокол, който се използва за предаване на информация за маршрутите между автономните системи.
- *Автономна система (autonomous system, AS)* — набор от маршрутизатори, притежаващи единни правила за маршрутизация, управлявани от една техническа администрация и работещи с един от IGP протоколите (за вътрешната маршрутизация в AS може да се използват и няколко IGP).
- *Транзитна автономна система (transit AS)* — автономна система, през която се предава трафика от други автономни системи.
- *Път (path)* — последователност, състояща се от номера на автономни системи, през които е необходимо да се премине, за да се достигне мрежата на получателя.
- *Атрибути на пътя (path attributes, PA)* — характеристики на пътя, които позволяват да се избере най-добрия път.
- *BGP speaker* — маршрутизатор, на който работи BGP протокол.
- *Съседи (neighbor, peer)* — всеки два маршрутизатора, между които има TCP съединение за обмен на информация за маршрутизацията.
- *Информация на мрежовия слой за достъпност на мрежата (Network Layer Reachability Information, NLRI)* — IP префикс и дължина на префикса.

Описание на протокола

BGP избира най-добрите маршрути не на основание на техническите характеристики на пътя (пропускателна способност, забавяния и т.н.), а на основание на политики. В локалните мрежи най-голямо значение има скоростта на сходимост на мрежата, времето за реагиране на измененията. Затова маршрутизаторите, които използват вътрешните протоколи за динамична маршрутизация, при избор на маршрута, като правило, сравняват някакви технически характеристики на пътя, например пропускателната способност на линиите.

При избор между каналите на два провайдера, често пъти има значение не кой канал е с по-добри технически характеристики, а някакви конкретни вътрешни правила на компанията. Например използването на кой канал ще бъде по-евтино за компанията. Затова за BGP изборът на най-добрия маршрут се осъществява на основание на политики, които се настройват с използането на филтри, с анонсирането на маршрути и с изменения на атрибутите.

Както и при другите протоколи за динамична маршрутизации, BGP може да предава трафик единствено на база на IP адреса на получателя. Това означава, че с помощта на BGP няма възможност за настройване на правилата за маршрутизация, които да се отчитат. Например това, от коя мрежа е бил изпратен пакета или данните за кое приложение да се предадат. Ако се взима решение за това как трябва да се маршрутизира пакета, тогава според някакви допълнителни критерии, освен адреса на получателя, е необходимо да се използва механизма policy-based routing (PBR).

Основни характеристики на протокола

BGP е path-vector протокол със следните характеристики:

- Използва TCP за предаване на данните, което осигурява надеждна доставка на обновленията на протокола (номер на порт - 179);
- Изпраща обновления само след изменения в мрежата (няма периодични обновления);
- Периодично изпраща keepalive съобщения за проверка на TCP съединенията;
- Метриката на протокола се нарича path vector или атрибути (attributes).

Автономна система

Автономната система (autonomous system, AS) това е система от IP мрежи и маршрутизатори, управлявани от един или няколко оператора, притежаваща единна, конкретно и ясно определена политика за маршрутизация в Интернет (RFC 1930).

Диапазони от номера на автономните системи (autonomous system number, ASN):

- 0-65535 (първоначално определен диапазон за ASN, 16 бита);
- 65536-4294967295 (нов диапазон за ASN, 32 бита (RFC 4893)).

Използване:

- 0 и 65535 (резервирани);
- 1-64495 (публични номера);
- 65552-4294967295 (допълнителни публични номера);
- 64512-65534 (частни номера);
- 23456 (представлява 32 битов диапазон на устройствата, които работят с 16 битов диапазон).

Описание на работата на протокола

- Таблица на съседите (neighbor table) — списък на всички BGP съседи;
- BGP таблица (BGP table, forwarding database, topology database);
    - Списък на мрежите, получени от всеки съсед;
    - Може да съдържа няколко пътя към destination мрежите;
    - Атрибути на BGP за всеки конкретен път.
- Таблица на маршрутизацията — списък на най-добрите пътища към мрежите.

По премълчаване BGP изпраща keepalive съобщения на всеки 60 секунди.

Ако съществуват няколко пътя до получателя, тогава маршрутизаторът ще анонсира на съседите не всички възможни варианти, а само най-добрия маршрут от BGP таблицата.

Вътрешен BGP (Internal BGP) и Външен BGP (External BGP)

- Вътрешен BGP (Internal BGP, iBGP) — BGP работи вътре в автономната система. iBGP съседите не е задължително да бъдат непосредствено съединени.
- Външен BGP (External BGP, eBGP) — BGP работи между автономните системи. По премълчаване, eBGP съседите са длъжни да бъдат непосредствено съединени.

Ако iBGP маршрутизаторите работят в нетранзитна AS, тогава съединението между тях трябва да бъде full mesh. Това е следствие от принципите на работа на протокола — ако маршрутизаторът, намиращ се на границата на AS, е получил обновление, тогава той го предава на всички съседи. Съседите, които се намират вътре в автономната система, повече това обновление не го разпространяват, понеже считат, че всички съседи вътре в AS вече са го получили.

Таймери на протокола

- Keepalive Interval — Интервал от време в секунди, между изпращанията на keepalive съобщенията. По премълчаване е 60 секунди.
- Hold Time — Интервал от време в секунди, след изтичането на който съседът ще се счита за недостъпен. По премълчаване е 180 секунди.

BGP съобщения

Полета на заглавието на протокола за всички типове BGP съобщения:

- Marker — поле, което е включено в заглавието за съвместимост. Размерът му е 16 байта.
- Length — дължина на цялото съобщение в октети, включвайки и заглавието. Полето може да приема стойности от 19 до 4096.
- Type — тип на предаваното съобщение:
    - 1 — OPEN

- 2 — UPDATE
- 3 — NOTIFICATION
- 4 — KEEPALIVE

Open

Open — използва се за установяване на отношения на съседство и за обмен на базови параметри. Изпраща се веднага след установяването на TCP съединението.

Update

Update — използва се за обмен на информация за маршрутизацията.

Notification

Notification — използва се когато възникват грешки в работата на BGP. След изпращането на съобщението сесията със съседа се прекратява.

Keepalive

Keepalive — използва се за поддържане на отношения на съседство, за откриване на неактивните съседи.

Отношения на съседство

За да се установят отношения на съседство в BGP, трябва да се настрои ръчно всеки съсед.

Когато се задава съсед на локалния маршрутизатор, задължително се посочва автономната система на съседа. Според тази информация BGP определя типа на съседа:

- Вътрешен BGP съсед (iBGP съсед) — съсед, който се намира в същата автономна система, в която е локалния маршрутизатор. iBGP съседите не е задължително да бъдат непосредствено съединени.
- Външен BGP съсед (eBGP съсед) — съсед, който се намира в автономна система, различна от тази, на локалния маршрутизатор. По премълчаване, eBGP съседите трябва да бъдат непосредствено съединени.

Типът на съседа малко влияе за установяване на отношенията на съседство. По-сериозни разлики между различните типове съседи се проявяват в процеса на изпращане на BGP обновления и при добавяне на маршрути в маршрутната таблица.

BGP изпълнява следните проверки, когато формира отношения на съседство:

1. Маршрутизаторът трябва да получи заявка за TCP съединение от адреса на изпращача - този, който маршрутизаторът открие в списъка на съседите (команда neighbor).
2. Номерът на автономната система на локалния маршрутизатор трябва да съвпада с номера на автономната система, която е посочена на съседния маршрутизатор с

командата *neighbor remote-as* (това изискване не се спазва при настройването на *конфедерации*).
3. Идентификаторите на маршрутизаторите (Router ID) не бива да съвпадат.
4. Ако е настроена автентификация, тогава съседите трябва да преминат през такава.

Първият пункт за проверка си има една особеност: само за единия от двата маршрутизатора IP адресът, зададен като адрес за изпращане на обновления, трябва да бъде посочен в командата *neighbor* на другия маршрутизатор.

BGP изпълнява проверка на таймерите keepalive и hold, но несъвпадението на тези параметри не влияе на установяването на отношенията на съседство. Ако таймерите не съвпадат, тогава всеки маршрутизатор ще използва по-малката стойност на таймера hold.

Състояние на връзката между съседите

- Idle
- Connect
- Open sent
- Open confirm
    - active
- Established

Табл. 3 **Етапи на формиране на съседство от BGP**

Състояние	Очакване на TCP съединение	Инициали-зация на TCP съеди-нение	Устано-вено ли е TCP съединение	Изпра-тено ли е Open съобщение	Полу-чено ли е Open съобщение	Съсе-дът в състояние Up ли се намира
Idle	Не					
Connect	Да					
Active	Да	Да				
Open sent	Да	Да	Да	Да		
Open confirm	Да	Да	Да	Да	Да	
Established	Да	Да	Да	Да	Да	Да

Ако не е съвпаднал IP адреса, с този на съседа, тогава същият съсед ще бъде в състояние active.

Атрибути на пътя (path attributes)

Атрибутите на пътя са разделени на 4 категории:

1. *Well-known* (общоизвестни) *mandatory* — всички маршрутизатори, работещи с BGP протокол, трябва да разпознават тези атрибути. Те задължително присъстват във всички обновления (update).
2. *Well-known discretionary* — всички маршрутизатори, работещи с BGP протокол, трябва да разпознават тези атрибути. Могат да присъстват в обновленията (update), но тяхното наличие не е задължително.
3. *Optional* (допълнителни) *transitive* — могат да не се разпознават от всички реализации на BGP. Ако маршрутизаторът не е разпознал атрибута, тогава той отбелязва обновлението като частично (partial) и го изпраща по-нататък на съседите, като съхранява неразпознатия атрибут.
4. *Optional non-transitive* — могат да не се разпознават от всички реализации на BGP. Ако маршрутизаторът не е разпознал атрибута, тогава атрибутът се игнорира и при предаване на съседите се отхвърля.

Примери за BGP атрибути:

- Well-known mandatory:
    - Autonomous system path
    - Next-hop
    - Origin
- Well-known discretionary:
    - Local preference
    - Atomic aggregate
- Optional transitive:
    - Aggregator
    - Communities
- Optional non-transitive:
    - Multi-exit discriminator (MED)
    - Originator ID
    - Cluster list

Autonomous system path

Атрибут Autonomous system path (AS Path):

- Описва през кои автономни системи трябва да се премине, за да се достигне до мрежата на получателя.
- Номерът на AS се добавя при предаването на обновления от една AS eBGP на съсед в друга AS.

Използва се за:

- Откриване на примки;
- Прилагане на политики.

Всеки сегмент с атрибут AS path е представен във вид на поле TLV (path segment type, path segment length, path segment value):

- *path segment type* — поле с размер 1 байт, за което са определени следните стойности:
    - 1 — AS_SET: неподредено множество от автономни системи, през които е преминал маршрутът в съобщението Update;
    - 2 — AS_SEQUENCE: подредено множество от автономни системи, през които е преминал маршрутът в съобщението Update;
- *path segment length* — поле с размер 1 байт. Задава колко автономни системи са посочени в полето path segment value;
- *path segment value* — номерата на автономните системи, всяка е представена от поле с размер 2 байта.

Next-hop

Атрибутът Next-hop:

- IP адрес на следващата AS за достигане на мрежата на получателя.
- Това е IP адресът на eBGP маршрутизатора, през който минава пътят към мрежата на получателя.
- Атрибутът се променя при предаването на префикса в друга AS.

Origin

Атрибутът Origin — определя по какъв начин е бил получен маршрутът в обновленията.

Възможни стойности на атрибута:

- 0 — IGP: NLRI е получено вътре в изходната автономна система;
- 1 — EGP: NLRI е научена от Exterior Gateway Protocol (EGP) протокол. Предшественик на BGP, който не се използва;
- 2 — Incomplete: NLRI е била научена по някакъв друг начин.

Local preference

Атрибутът Local preference:

- Показва на маршрутизаторите вътре в автономната система как да излязат навън от нейните граници.
- Този атрибут се предава само в рамките на една автономна система.
- На Cisco маршрутизаторите по премълчаване стойността на този атрибут е 100.
- Избира се тази точка за изход, на която стойността на атрибута е по-голяма.
- Ако eBGP съсед получава обновление със зададена стойност на local preference, тогава той игнорира този атрибут.

Atomic aggregate

Етикет, показващ, че NLRI е summary.

Aggregator

Списъкът от RID и ASN маршрутизатори, създали summary NLRI.

Communities

Атрибутът community задава:

- Тагиране на маршрути;
- Съществуване на предопределени стойности;
- По премълчаване не се препраща на съседи;
- Един от вариантите за приложение: се предава на съседната AS за управление на входящия трафик.

Стойностите от 0x00000000 до 0x0000FFFF и от 0xFFFF0000 до 0xFFFFFFFF са резервирани.

Като правило community се изобразява във формат ASN:VALUE. В такъв формат са достъпни за използване стойностите за community от 1:0 до 65534:65535. В първата част се задава номерът на автономната система, а във втората - стойността community, което определя политиката за маршрутизация на трафика.

Някои стойности на communities са предопределени. RFC1997 определя три такива стойности за community. Те трябва еднакво да се разпознават и обработват от всички реализации на BGP, които разпознават атрибута community.

Ако маршрутизаторът получава маршрут, в който е посочена предопределена стойност за communities, тогава той изпълнява специфични, предопределени действия, базирани на стойността на атрибута.

Предопределени стойности за communities (Well-known Communities):

- *no-export (0xFFFFFF01)* — Всички маршрути, които се предават с такава стойност на атрибута community, не бива да се анонсират извън границите на конфедерацията (автономна система, която не е част от конфедерация се счита за конфедерация). Т. е., маршрутите не се анонсират на EBGP съседите, но се анонсират на външни съседи в конфедерацията;
- *no-advertise (0xFFFFFF02)* — Всички маршрути, които се предават с такива стойности на атрибута community, не бива да се анонсират на други BGP съседи;
- *no-export-subconfed (0xFFFFFF03)* — Всички маршрути, които се предават с такива стойности на атрибута community, не бива да се анонсират на външни BGP съседи (нито на външни в конфедерацията, нито на настоящи външни съседи). В Cisco тази стойност се среща и под наименованието local-as.

Маршрутизаторите, които не поддържат атрибута community, ще го предават по-нататък, понеже той е transitive атрибут.

Multi exit discriminator (MED)

Атрибут MED:

- Използва се за информиране на eBGP съседите, какъв път в автономната система е за предпочитане.
- Атрибутът се предава между автономните системи.
- Маршрутизаторите вътре в съседната автономна система използват този атрибут, но когато обновлението излиза извън AS, атрибутът MED се отхвърля.
- Колкото е по-малка стойността на атрибута, толкова е по-предпочитана за точка на вход в автономната система.

Weight атрибут (от Cisco)

Атрибут Weight:

- Позволява да се задава "тегло" на различните пътища, локално на маршрутизатора.
- Използва се в тези случаи, когато един маршрутизатор има няколко изхода от автономната система (сам маршрутизаторът се явява точка за изход).
- Има смисъл само локално, в границите на маршрутизатора.
- Не се предава в обновленията.
- Колкото е по-голяма стойността на атрибута, толкова той е по-предпочитан за път за изход.

Избор на пътя

Характеристики на процедурата за избор на пътя от BGP протокола:

- В BGP таблицата се съхраняват всички известни пътища, а в маршрутната таблица — най-добрите.
- Пътищата се избират на база на политики.
- Пътищата не се избират на основание на пропускателна способност.

Първо се проверява:

- Достъпен ли е next-hop (Route Resolvability Condition)?

За да се счита next-hop за достъпен (accessible), е необходимо в маршрутната таблица да има IGP маршрут, който до води към него.

Cisco

На Cisco маршрутизатор, ако не са настроени никакви политики за избор на път, изборът на път става според следния алгоритъм (като на всяка следваща стъпка маршрутизаторът преминава само при съвпадение на стойностите от предишната стъпка):

1. Максималната стойност за weight (локално за маршрутизатора).
2. Максималната стойност за local preference (за цялата AS).
3. Да се предпочете локалният маршрут на маршрутизатора (next hop = 0.0.0.0).
4. Най-краткият път през автономните системи (най-краткия AS_PATH).
5. Минималната стойност за origin code (IGP < EGP < incomplete).
6. Минималната стойност за MED (разпространява се между автономните системи).
7. Пътят eBGP е по-добър от път iBGP.
8. Да се избере път през най-близкия IGP съсед.
9. Да се избере най-старият маршрут за eBGP пътя.
10. Да се избере път през съсед с най-малко BGP router ID.
11. Да се избере път през съсед с най-малък IP адрес.

Juniper

Ако съществуват няколко маршрута до една мрежа цел, ще бъде избран само един от тях. Всяка стъпка в алгоритъма за избор на най-добрия маршрут се опитва да отстрани всички, освен един от маршрутите към пункта на целта. Ако на дадена стъпка на алгоритъма маршрутите все още са повече от един, тогава се изпълнява преход към следващата стъпка от алгоритъма. Така алгоритъмът работи до тогава, докато това е необходимо. В Juniper устройствата изборът на най-добрия маршрут се реализира според следния алгоритъм:

1. Проверка за достъпност next-hop в локалната маршрутна таблица. Ако next-hop не е достъпен, маршрутът се отхвърля.
2. Маршрутизаторът избира маршрут с най-голям Local Preference атрибут.
3. Маршрутизаторът избира маршрут с най-малка стойност на AS Path length.

4. Маршрутизаторът избира маршрут с най-малка стойност на атрибута Origin (т. е. се предпочита IGP).
5. Маршрутизаторът избира маршрут с най-малка стойност за MED. Тази стъпка се изпълнява, по премълчаване, само за маршрути в една AS.
6. Маршрутизаторът избира маршрути, получени от EBGP съседи, а не такива, получени от IBGP соседи. Ако останалите маршрути са EBGP маршрути, маршрутизаторът преминава към стъпка 9.
7. Маршрутизаторът избира маршрут с най-малка метрика IGP за анонсируемия BGP Next Hop.
8. Ако се използва Route Reflection за IBGP пиринг, маршрутизаторът избира път с най-малка стойност на Cluster-List length.
9. Маршрутизаторът избира маршрут от партньор с най-малко Router ID.
10. Маршрутизаторът избира маршрут от партньор с най-малък Peer Address.

Само най-добрият път се записва в маршрутната таблица и се анонсира на BGP съседите.

16. Виртуални локални мрежи

Виртуалните мрежи са създадени, за да реализират сегментация на мрежата с комутатори. В традиционните мрежи делението на броудкастни домейни се осъществява от маршрутизатори. VLAN се явява един от основните методи за защита на информацията в мрежите с комутатори.

Виртуалната локална мрежа VLAN се състои от възли, обединени в общодостъпен (броудкастен) домейн, образуван от присъединени към виртуалната мрежа портове на комутатора.

Виртуалните локални мрежи логически сегментират цялата мрежа на общодостъпни домейни така, че пакетите да се прехвърлят само между портовете, които принадлежат на една конкретна VLAN.

Трафикът между различните VLAN се осигурява чрез маршрутизация, т. е. комуникацията между възлите на различните виртуални мрежи се реализира задължително с помощта на маршрутизатор или комутатор на трети слой (комуникацията между възлите на различните виртуални мрежи се реализира само с маршрутизатор).

На практика се използват няколко типа виртуални локални мрежи: VLAN за предаване на данни; VLAN за предаване на управляващ трафик; VLAN за предаване на гласов трафик.

По подразбиране всички портове на комутатора са предназначени за първата виртуална локална мрежа VLAN1 и се използват за предаване на данни.

За управлението на виртуалните локални мрежи, в това число и на отдалечен достъп, е определен виртуалния интерфейс SVI (един от VLAN, по подразбиране VLAN1). На този интерфейс се задават IP адрес, мрежова маска и шлюз по подразбиране. Администраторите обикновено променят номера на управляващата мрежа с цел повишаване на безопасността.

На всяка виртуална мрежа при конфигурирането трябва да се зададе IP адрес на мрежата или на подмрежата със съответстващи маска и шлюз.

Функционирането на виртуалните локални мрежи се определя от 802.1Q протокол.

При построяването на мрежи с няколко комутатора, в заглавието на кадъра се добавя уникален идентификатор - таг (tag) на виртуалната мрежа, който определя членството на VLAN за всеки пакет.

Маркирането (тагирането) се използва при обмен на данните от VLAN мрежите между комутаторите.

Тагът с размер 2 байта се слага в кадъра между полето адрес на източника и полето Тип/Дължина. 12 бита за тага (VLAN ID) се използват за идентификация на VLAN, което позволява да се маркира (тагира) до 4095 виртуални мрежи (като стойностите 0, 1 и 4095 са резервирани), което обхваща нормален (1 - 1005) и разширен (1006 - 4094) диапазони на идентификаторите на VLAN.

Съвкупността от физически канали между две устройства може да бъде заменена от един агрегиран логически канал, получил наименованието транк (Trunk).

Транк се нарича магистрален канал, предаващ кадри на няколко виртуални локални мрежи.

Транковите портове предават тагиран трафик на няколко VLAN мрежи. Портовете за достъп предават нетагиран трафик на една VLAN мрежа.

От 802.1Q протокола е предвидена собствена мрежа native VLAN, която е предназначена за транковия порт, но трафика на която се предава нетагиран.

Съгласуването на магистралните транкови канали се реализира динамично от транковия протокол DTP, който е протокол на Cisco, и не поддържи оборудването на другите производители. За немагистралните канали се препоръчва да се изключва DTP протокола.

Защитените портове забраняват обмена на данни между интерфейсите при всяко препращане. При това се създава гранична частна мрежа PVLAN.

За осигуряването на маршрутизация между VLAN при използването на транкови съединения на интерфейса на маршрутизатора се формират няколко субинтерфейса, според броя на виртуалните локални мрежи.

17. Транспортен слой

Транспортният слой (transport layer) е най-интересен за администраторите и разработчиците на мрежи, понеже той управлява предаването на съобщенията между крайните възли в мрежата от край до край ("end-end"), като осигурява надеждност и икономическа ефективност на предаването на данните независимо от потребителя. При това крайните възли е възможно да взаимодействат през няколко междинни възела или даже през няколко транзитни мрежи.

На транспортния слой се реализират:

1. преобразуване на дългите съобщения в пакети при предаването им в мрежата и обратното преобразуване;
2. контрол на последователността на преминаването на пакетите;
3. регулиране на трафика в мрежата;
4. разпознаване на дублираните пакети и тяхното унищожаване.

Комуникацията "end-end" разполага с още един метод за адресация – адрес на процеса, който се съпоставя с определена приложна програма (приложен процес), изпълнявана на компютъра. Компютърът обичайно изпълнява едновременно няколко програми, във връзка с което е необходимо да се знае за коя приложна програма (процес) е предназначено постъпилото съобщение. За това на транспортния слой се използва специален адрес, наричан адрес на порта (port number). Мрежовият слой доставя всеки пакет на конкретен адрес на компютъра (IPaddr), а транспортният слой предава напълно събраното съобщение на конкретния приложен процес на този компютър (port №).

Задачата на IP е предаване на данни между двойка мрежови интерфейси. Задачата на транспортния слой е предаване на данни между двойка приложни процеси, изпълнявани в мрежата. С други думи IP предава данни между два различни MAC адреса; а TCP и UDP - между двойка приложни процеси.

Според номера на порта транспортните протоколи определят на кое приложение трябва да се предаде съдържимото на пакетите. Съвкупността от IP адрес и номер на порт се нарича *сокет*. Сокетът уникално идентифицира приложния процес в Интернет.

Транспортният слой може да предоставя различни типове услуги, в частност предаване на данни без установяване на съединение или с предварително установяване на съединението. В последния случай преди началото на предаването на данните посредством използването на специални управляващи пакети се установява съединение с транспортния слой на компютъра, за когото са предназначени предаваните данни. След като се предадат всички данни, съединението се затваря. При предаването на данни без установяване на съединение, транспортният слой се използва за предаването на единични пакети, наричани дейтаграми, без да се гарантира тяхната надеждна доставка. Предаването на данни с установяване на съединение се прилага за надеждна доставка на данните.

18. Transmission Control Protocol

TCP (протоколът за управление на предаването) е протокол от транспортния и сеансовия слой на OSI/RM модела.

Полета на TCP сегмента:

- Source Port – порт на източника (2 байта);
- Destination Port – порт на получателя (2 байта);
- Sequence Number – номер на последователността (4 байта);
- Acknowledgement Number – номер на потвърждението (4 байта), като последните две стойности се използват за контрол на правилността на получените данни;
- Data Offset – дължина на TCP заглавието в 32 битови думи (4 бита);
- еднобитови TCP флагове (U, A, P, R, S, F);
- Window – размер на прозореца в байтове (2 байта);
- Checksum – контролна сума на TCP сегмента и на псевдозаглавието (2 байта);
- Urgent pointer – указател за спешност (2 байта);
- Option-kind, Option-length и Option-data – тип (1 байт), дължина (1 байт) и данни за опциите на TCP;
- Padding – поле за запълване до 32 битова дума;
- Data – данните от горния слой, инкапсулирани в TCP сегмент.
- TCP флагове:
  - URG (urgent) – флаг за спешност, задава за използване на полето Urgent Pointer (например при натискане на Ctrl-C в сеанса на telnet или при предаването на файл в полето Urgent Pointer се записва отместването на предаваните в този сегмент данни относно номера на последователността);
  - ACK (acknowledgement) – флаг за потвърждаване, задава потвърждаване на приемането на данните и необходимостта от използването на Acknowledgement Number;
  - PSH (push) – указател за незабавно предаване на инкапсулирани данни за приложението на страната на получателя (например в сеанса на telnet за изпращането на сегменти, съдържащи отделни букви, при побайтово изпращане за сървъра);
  - RST (reset) – флаг за отхвърляне на съединението;
  - SYN (synchronization) – флаг за инициализация на съединението, посочва необходимостта от използването на Sequence Number;
  - FIN (finish) – флаг за заявка за затваряне на съединението.

TCP протоколът осигурява надеждно предаване на данните между приложните процеси, понеже използва установяване на логическо съединение между взаимодействащите процеси.

Логическото съединение между два приложни процеса се идентифицира от двойка сокети (IP адрес, номер на порт), всеки от които описва един от взаимодействащите процеси.

Информацията, постъпваща към TCP протокол в рамките на логическото съединение от протоколите от по-горния слой, се разглежда от TCP протокола като неструктуриран поток от байтове и се записва в буфер. За предаването към мрежовия слой от буфера се изрязва сегмент, непревишаващ 64 Кбайта (максималния размера на IP пакет). На практика дължината на сегмента се ограничива от стойността 1460 байта, като така се осигурява той да се разположи в Ethernet кадър с TCP и IP заглавия.

TCP съединението е ориентирано към пълно дуплексно предаване. Управлението на потока от данни в TCP протокола се осъществява с използването на механизма на плаващия прозорец с променлив размер. При предаването на сегмента възелът изпращач включва таймер и очаква потвърждение. Отрицателни квитанции не се изпращат, а се използва механизмът за таймаут. Възелът цел като получи сегмента формира и изпраща обратно сегмент (с данни, ако има такива, или без данни) с номер за потвърждение, равен на следващия пореден номер на очаквания байт. За разлика от много други протоколи, TCP протоколът потвърждава получаването не на пакети, а на байтове от потока. Ако времето за очакване на потвърждението изтече, изпращачът изпраща сегмента отново (повторно).

Въпреки че изглежда прост в работата си протокол, в него има редица нюанси, които могат да доведат до някои проблеми. Първо, понеже сегментите при предаването по мрежата могат да се фрагментират, е възможна ситуация, при която част от предадения сегмент ще бъде приета, а останалата част ще се окаже изгубена. Второ, сегментите могат да пристигат във възела на получателя в произволен ред, което може да доведе до ситуация, при която байтовете от 2345 до 3456 вече са получени, но потвърждението за тях не може да бъде изпратено, понеже байтовете от 1234 до 2344 все още не са получени. Трето, сегментите могат да се задържат в мрежата толкова дълго, че при изпращача да изтече интервала за очакване на потвърждението и той да ги изпрати отново. Предаденият повторно сегмент може да премине по друг маршрут и може да бъде фрагментиран по друг начин, или сегментът може по маршрута да попадне случайно в претоварена мрежа. Като резултат за възстановяването на изходния сегмент ще се изисква достатъчно сложна обработка.

TCP осигурява своята надеждност чрез използването на следните механизми:

- Данните от приложението се разбиват на блокове (сегменти) с определен размер, които ще се изпращат.
- Когато TCP изпраща сегмент, той устанавява таймер, очаквайки, че от отдалечената страна ще пристигне потвърждение за този сегмент, че е получен. Ако потвърждението не е получено до изтичането на времето, сегментът се изпраща повторно.
- Когато TCP приема данните от отдалечената страна на съединението, то изпраща потвърждение. Това потвърждение не се изпраща незабавно, а обикновено след части от секундата.
- TCP осъществява изчисляване на контролна сума за своето заглавие и данните. Тази контролна сума се изчислява и в двата края на съединението, като целта й е да се открият всякакви изменения в данните по време на предаването им. Ако сегментът е с невярна контролна сума, TCP го отхвърля и потвърждение не се генерира. (Очаква се, че изпращачът ще отработи таймаута и ще направи повторно предаване.)

- Понеже TCP сегментите се инкапсулират във вид на IP дейтаграми, а IP дейтаграмите могат да са с нарушена последователност при предаването, също така и TCP сегментите могат да са с нарушена последователност при предаването. След получаването на данните TCP може при необходимост да измени тяхната последователност, като резултат приложението получава данните в правилен ред.
- Понеже IP дейтаграмата може да се дублира, приемащата страна на TCP трябва да отхвърли дублираните сегменти.
- TCP осъществява контрол на потока от данни. Всяка страна на TCP съединението има определено буферно пространство. TCP на приемащата страна позволява на отдалечената страна да изпраща данни само в този случай, ако получателят може да ги разположи в буфера. Последното предотвратява препълването на буферите на бавни хостове от бързи хостове.

Инкапсулация на TCP

TCP е байт-ориентиран (byte-stream) протокол. TCP протоколът разглежда данните на клиента и на сървъра като непрекъснат неинтерпретируем поток от октети. TCP разделя този поток на части за прехвърляне към друг възел в TCP сегменти с някакъв размер. За изпращането или получаването на сегмента TCP модулът се обръща към IP модула.

TCP сегментите се инкапсулират в IP дейтаграми и могат да съдържат параметри на TCP. За TCP протокола е определен максималния размер на сегмента (maximum segment size, MSS). MSS посочва на събеседника максималния обем от данни, който може да изпраща във всеки сегмент на TCP. Целта на параметъра MSS е да съобщи на събеседника реалния размер на буфера за получаването (събирането) и да се опита да предотврати фрагментацията от IP модула. Този параметър на TCP позволява на възела, изпращащ сегмент за синхронизация SYN, да обяви на събеседника си своя MSS - максималното количество данни, което той ще приема с всеки TCP сегмент по това съединение. В качеството на MSS често пъти се използва стойност, равна на MTU на интерфейса минус фиксираните размери на заглавията на IP и на TCP. В локална мрежа Ethernet при използването на IPv4 максималният размер на сегмента е 1460 байта.

Стойността на MSS в TCP пакета е представена в 16 разрядно поле, ограничаващо стойността до 65535. Това е допустимо за IPv4, понеже максималното количество данни на TCP в дейтаграмата на IPv4 е равно на 65495 байта (65 535 минус 20 байтово заглавие на IPv4 и минус 20 байтово заглавие на TCP). Стойността на MSS, равна на 65535, се счита за особен случай, задаващ «безкрайност». Тази стойност се използва само заедно с параметъра на прозореца за увеличаване на обема на полезните данни, когато се изисква размер на MTU, превишаващ 65535. Ако TCP използва увеличаване на обема на полезните данни и получава от събеседника обявление за размера на MSS, равен на 65535 байта, тогава максималният размер на дейтаграмата, изпращана от него, ще бъде равен на MTU на интерфейса. Ако се окаже, че този размер е твърде голям (например по пътя съществува канал с по-малък размер на MTU), тогава ще бъде установена по-малка стойност.

Максималният размер на прозореца, който може да бъде зададен в заглавието на TCP, е равен на 65535. Но високоскоростните съединения (45 Mbps и повече) или линии

с големи забавяния (спътникови мрежи) изискват по-голям размер на прозореца за достигане на максимално възможна пропускателна способност. Параметърът на TCP за мащабиране на прозорец (Window scale option) определя, че обявената в заглавието на TCP величина на прозореца, трябва да се мащабира – да се изместят наляво от 0 до 14 разряда (бита), като така се предоставя максимално възможен прозорец с размер почти на гигабайт (65 535 x 2^14). За да може да се използва този параметър в съединението е необходима неговата поддръжка от двата крайни възела.

Управление на потока

TCP осигурява управление на потока (flow control). TCP винаги съобщава на своя събеседник колко конкретно байта той иска да получи от него. Това се нарича обявяване на прозореца (window). За всеки момент от време прозорецът съответства на свободното пространство в приемащия буфер. Така се гарантира, че изпращачът няма да препълни буфера на получателя. Прозорецът се променя динамично с течение на времето: според това с какви обеми пристигат данните от изпращача, размерът на прозореца се намалява, но според размера на прочитането на данните от приемащото приложение, прозорецът се увеличива. Прозорецът може да се нулира. Например, ако приемащият буфер на TCP за дадения сокет се запълни, той трябва да изчака, докато приложението прочете данните от буфера, преди да може да започне да получава други данни.

TCP прилага алгоритми, позволяващи динамично да се прогнозира времето (периода) на оборота (round-trip time, RTT) на пакетите между клиента и сървъра и така да се изчислява колко време е необходимо за получаването на потвърждението. Например RTT в локална мрежа може да има стойност от порядъка на милисекунди, а в същото време за глобална мрежа тази величина може да достига няколко секунди. Освен това в определен момент от време TCP може да получит стойност за RTT между конкретни клиент и сървър, равна на една секунда, а после след 30 секунди да се измери RTT за същото съединение и да се получи стойност, равна на няколко секунди, което се обяснява с разликите в мрежовия трафик.

TCP съединение

Съединението е съвкупност от информация за състоянието на потока от данни, включваща сокетите, номерата на изпратените, приетите и потвърдените октети, размерите на прозорците.

Всяко съединение уникално се идентифицира в Интернет от двойката сокети.

Съединението се характеризира за процеса с име, което представлява указател към TCB (Transmission Control Block) структура, съдържаща информация за съединението.

Отварянето на съединението от процеса се реализира с извикването на функцията OPEN, на която се предава сокета, с която трябва да се установи съединението.

Функцията връща името на съединението. Различават се два типа отваряне на съединението: активно и пасивно.

Работата с TCP протокол между двама абонати винаги преминава през следните три етапа: установяване (отваряне) на съединение, обмен на данни и затваряне на съединението.

Отваряне и затваряне на TCP съединение

За да се отвори (установи) TCP съединение, е необходимо да се извършат следните действия.

1. Страната - инициатор на съединението (клиент) изпраща SYN сегмент (вдигнат бит SYN в полето за флагове на заглавието на TCP), посочвайки името на домейна (или IPdst) и номера на порта на сървъра, към който клиентът иска да се присъедини, и началния номер на последователността на клиента (поле Sequence Number в заглавието на TCP).
2. Сървърът отговаря със сегмент SYN, съдържащ своя начален номер на последователността на сървъра заедно с вдигнат флаг SYN. Също така е вдигнат флаг ACK и е попълнено полето "номер на потвърждението" (Acknowledgement number), където се записва полученият от клиента номер на последователността + 1.
3. Клиентът трябва да потвърди пристигането на SYN сегмента от сървъра с използването на ACK флага и с нова стойност в полето за потвърждение (полученият от сървъра номер на последователността +1).

Предаването на тези три сегмента е достатъчно за установяването на съединението (често този процес се нарича три стъпково ръкостискане, three - way handshaking). След това между страните (клиента и сървъра) е възможен двустранен обмен на данни по установеното съединение.

При едностранно затваряне на съединението страната - инициатор на затварянето трябва да изпрати по установеното съединение FIN сегмент (с вдигнат флаг FIN), а също така и да получи ACK отговор от отдалечената страна с уведомление за получаването на FIN сегмента.

След това съединението с възможност за двустранен обмен на данни преминава в еднопосочно състояние (едната страна е затворила съединението, а втората е активна и поддържа отворено съединението). За да се затвори напълно съединението, активната страна трябва да формира FIN сегмент и да получи за него потвърждение.

Така при установяването на съединението на двете страни е необходимо да изпратят и да приемат 3 сегмента, а при затварянето му – 4.

Междинни състояния на TCP съединението

TCP съединението по време на функциониране преминава през редица междинни състояния. Това са състоянията LISTEN, SYN-SENT, SYN-RECEIVED, ESTABLISHED, FIN-WAIT-1, FIN-WAIT-2, CLOSE-WAIT, CLOSING, LAST-ACK, TIME-WAIT, а също така и фиктивното състояние CLOSED (състоянието CLOSED е фиктивно, понеже то представлява отсътствие на съединение). Преходът от едно състояние в друго (следващо) се реализира като отговор на събитие, например: заявки на клиента, пристигане на сегментите, изтичане на контролното време.

Определени са следните заявки на процеса на клиента в TCP модула (с всяка заявка, освен OPEN, се предава името на съединението):

ACTIVE-OPEN – активно отваряне на съединението;

PASSIVE-OPEN – пасивно отваряне на съединението;

SEND – изпращане на данни (предава се указател към буфер с данни, размера на буфера, стойности на флаговете URG и PSH);

RECEIVE – получаване на данните (предава се указател към буфер с данни, размера на буфера; връща се броячът на получените октети, стойностите на флаговете URG и PSH);

STATUS – запитване за състоянието на съединението;

CLOSE – затваряне на съединението (изпращат се всички недоизпратени данни и се обменят сегменти с бит FIN);

ABORT – ликвидация на съединението (унищожават се блока TCB и всички недоизпратени данни, изпраща се сегмент с вдигнат RST бит).

LISTEN – локалният процес пасивно очаква заявки от страна на отдалечени сокети.

SYN-SENT – локалният процес е изпратил свой SYN и очаква отдалечен SYN.

SYN-RECEIVED – локалният процес е получил отдалечен SYN, изпратил е (по-рано или току-що) свой SYN и очаква ACK на свой SYN.

ESTABLISHED – локалният процес е изпратил ACK на отдалечен SYN, получил е ACK на свой SYN; съединението е установено.

FIN-WAIT-1 – локалният процес пръв е изпратил свой FIN и очаква реакция на отсрещната страна; при това е възможно той да продължава да получава данни.

FIN-WAIT-2 – локалният процес е получил ACK за свой по-рано изпратен FIN, но не е получил от отдалечената страна FIN; очаква този FIN; при което е възможно той да продължава да получава данни.

CLOSE-WAIT – локалният процес не е изпратил свой FIN (възможно е да няма намерение да прекратява съединението), получава от отдалечената страна FIN; изпраща ACK на този FIN, но при това е възможно той да продължава да изпраща данни.

LAST-ACK – процесът е изпратил свой FIN, но по-рано той вече е получил FIN от отдалечената страна и е изпратил за него ACK; затова процесът очаква да получи ACK на своя FIN, за да затвори окончателно съединението.

CLOSING – локалният процес вече е изпратил свой FIN и все още не е получил за него потвърждение, но е получил отдалечен FIN (и е изпратил за него ACK); очаква ACK на своя FIN.

TIME-WAIT – локалният процес е изпратил по-рано свой FIN и е получил за него потвърждение, получил е FIN от отдалечената страна и току-що е изпратил ACK за него; сега този процес очаква известно време (две времена на живот на сегмента, обикновено 4 минути) за гаранция на това, че отсрещната страна ще получи неговия ACK за своя FIN, след което съединението ще се затвори окончателно.

CLOSED – няма съединение.

Дейността на програмния модул на TCP протокол може да се разглежда като реагиране на събития в зависимост от състоянието на съединение.

Алгоритъм на «плаващия прозорец»

Особеност на работата: единицата на предаваните данни е сегментът.

Размерът на плаващия прозорец се измерва в байтове и той може да се променя по време на работа на TCP протокола.

При установяването на TCP съединението не е ясно каква е пропускателната способност на канала и какъв размер на прозореца е необходимо да се избере. Затова се използва следния алгоритъм:

1. Размерът на прозореца се установява като максимален, а после се намалява по време на работа.
3. Формулата за определяне размера на прозореца: при всяко предаване на пакета се изчислява времето на оборота на пакета, т.е. времето за достигането на пакета от единия адресат до другия и назад (RTT). Тази величина се определя постоянно. При изпращането на всеки сегмент това време се измерва самостоятелно.
4. Размерът на прозореца се определя като средно претеглено според последните 10 натрупани стойности (или използва 10 такива). На тях се задават коефициенти от 1 до 10 (най-голям коефициент има последната стойност). Тези стойности се сумират и се изчислява средното аритметично, като най-голямо тегло ще имат последните стойности на размера на прозореца. Така се определя колко данни могат да се предадат за това време (обемът на предаваните данни за единица време).

Понеже това изчисление се извършва постоянно, формулата позволява динамично да се адаптира размерът на прозореца според пропускателната способност.

Например ако пропускателната способност рязко се намали, обемът на данните, предавани без потвърждение, ще се намали.

Частни случаи: да предположим, че изпращачът предава данните, а на страната на получателя на данните пристигат в буфера на TCP, но приложната програма не се обръща към тези данни. Реализира се препълване на буфера.

TCP модулът постъпва по следния начин: той започва да изпраща потвърждения на последния приет байт и установява размер на прозореца 0. Това се нарича «зондиране с празен прозорец».

Резултатът е: изпращачът периодично получава потвърждение на един и същи байт, понеже той получава потвърждение (няма тишина в канала), изпращачът знае, че потребителят е все още «жив». Нулевият размер на прозореца не му позволява да изпраща данни, т.е. изпращачът също спира предаването.

19. User Datagram Protocol

UDP (протоколът за предаване на потребителски дейтаграми) е протокол от транспортния слой на OSI/RM модела.

Полета на UDP дайтаграмата:

- Source Port – порт на източника (2 байта);
- Destination Port – порт на получателя (2 байта);
- Length – пълна дължина на UDP пакета (2 байта);
- Checksum – контролна сума (2 байта);
- Data – данни, инкапсулирани в UDP пакет.

UDP протоколът не осигурява надеждно предаване на пакетите. UDP не прилага такива механизми, като: потвърждения, поредни номера, определяне на RTT, таймаути или повторно предаване. Ако UDP дейтаграма се дублира в мрежата, тогава възелът на получателя може да получи и двата екземпляра. Също така, ако UDP клиент изпрати две дейтаграми на един и същ получател, тяхната последователност може да се промени по време на получаването, и те ще бъдат доставени с нарушение на изходния ред. UDP приложенията са длъжни да обработват всички подобни случаи.

UDP не осигурява управление на потока. Бърз изпращач на UDP пакети може лесно да предава дейтаграми с такава скорост, с която да не може да работи получателя им.

Основни разлики спрямо TCP:

- Отсъства съединение между UDP модулите;
- Съобщението не се разбива при предаването му;
- При загуба на пакет не се изпраща заявка за повторно предаване.

UDP се използва, когато не се изисква гарантирана доставка на пакетите, например за потоково видео и аудио, DNS (понеже данните са с неголям размер). Ако при проверката на контролната сума се регистрира грешка или ако процесът изисква свързане към номер на порт, който не съществува, пакетът се унищожава. Ако пакетите постъпват по-бързо, отколкото UDP модулът успява да ги обработва, тогава постъпващите пакети се игнорират.

Не всички полета на UDP пакета задължително трябва да са запълнени. Ако изпращаната дейтаграма не предполага получаване на отговор, тогава на мястото на адреса на изпращача могат да се запишат нули.

20. Команди за тестване на мрежата

ping

Програмата ping е предназначена за проверка на достижимостта на отдалечен хост или мрежова услуга и често пъти се използва в качеството на първична диагностика за неизправности (отсъствия на връзка) в глобалните и в локалните мрежи. Програмата изпраща ICMP (Internet Control Message Protocol) exo заявка към хост и очаква получаване (връщане) на ICMP exo отговор. Времето между изпращането на ICMP заявка и получаването на отговора се нарича round-trip time (RTT) и то може да оценява пропускателната способност на свързващата линия.

Програмата ping е реализирана във всички мрежови операционни системи. С нейна помощ може да се тества мрежа с IP пакети с всякаква дължина, от минималната (64 байта) до максимално допустимата (~64 килобайта). Дължината на пакета се задава опционално в командния ред.

Следва да се отчита, че често пъти ping пакетите могат да се блокират от междинни маршрутизатори. Също така и хостовете могат да бъдат конфигурирани да забраняват приемането / изпращането на ICMP съобщения.

Програмата ping представя резултатите от измерването на RTT, където се показват минималната / средната / максималната стойност за няколко пакета и отклонението от средното. Това време е сума от времето за преминаване до целта и обратно и времето за обработка на пакета на отдачената страна и от междинните маршрутизатори (се измерва в микросекунди).

Може да се определи пропускателната способност на мрежата, като се построи графика на зависимостта на RTT от дължината на ICMP съобщението.

При изпращане на ICMP пакет с дължина по-голяма от 1.5 килобайта в Ethernet мрежа той ще се фрагментира.

traceroute (tracert)

Програмата traceroute позволява да се прегледа маршрута, по който се движат IP дейтаграмите от един хост до друг. Обикновено две последователни дейтаграми, изпратени от един и същ източник до един и същ получател, преминават по еднакъв маршрут, обаче да се гарантира последното е невъзможно. С помощта на traceroute може да се приложи IP опцията маршрутизация от източника (*source routing)*.

traceroute, както и ping, използва ICMP. Също така и полето TTL в IP заглавието.

Полето TTL (време на живот) е 8 битово поле, в което изпращачът записва конкретна стойност (или по подразбиране, която зависи от операционната система).

Всеки маршрутизатор, който обработва дейтаграмата, намалява стойността на TTL с единица или на броя секунди, в течение на които маршрутизаторът е обработвал дейтаграмата. Понеже повечето маршрутизатори задържат дейтаграмата за по-малко от секунда, полето TTL, като правило, се намалява с единица и достатъчно точно съответства на броя на ретранслациите (forward).

За хоста получател се изпраща IP (или UDP при използване на DNS) дейтаграма 1 с TTL, зададено като единица. Първият маршрутизатор, който трябва да обработи дейтаграма 1, я унищожава (понеже TTL е равно на 1) и изпраща ICMP съобщение 2 за изтичане на времето (time exceeded).

ICMP съобщението е инкапсулирано в IP дейтаграма 2, и следователно, изпращачът на първоначалната UDP дейтаграма 1 знае IP адреса на изпращач 2. Така се определя първия маршрутизатор, през който се достига хоста получател.

След това traceroute изпраща дейтаграма с TTL, равно на 2, което позволява да се получи IP адреса на втория маршрутизатор. Това продължава до тогава, докато дейтаграмата не достигне хоста получател. Ако дейтаграмата достигне хоста получател, той няма да я унищожи и няма да генерира ICMP съобщение за изтичане на времето, понеже дейтаграмата е достигнала своя краен пункт за получаване.

Резултатът от изпълнението на командата може да съдържа редове със звездички. Те говорят за това, че пакетът или се е изгубил (е отхвърлен вследствие на претоварване на маршрутизаторите), или маршрутизаторът няма права да изпраща ICMP съобщение "time exceeded" (за случая с * * *).

netstat

Програмата netstat показва текущото състояние на различните структури от данни, свързани с мрежовото взаимодействие. По подразбиране извежда списък на отворени сокети. В зависимост от избраните опции може да се изведе информация за маршрутната таблица (-r), всички отворени сокети, без значение дали слушат за устновяване на съединение, включително показва и UDP сокети (-a), само TCP сокети (-t), статистическа информация за протоколите (-s).

За TCP сокетите (преглед с опция -a) са допустими следните състояния:

CLOSED – затворено, сокетът не се използва;

LISTEN – очакване на входящи съединения;

SYN_SENT – опит за установяване на съединение;

SYN_RECEIVED – процес на начална синхронизация на съединението;

ESTABLISHED – съединението е установено;

CLOSE_WAIT – отдалечената страна се е изключила; очаква се затваряне на сокета;

FIN_WAIT_1 – сокетът е затворен; изключване на съединението (изпращане на флаг FIN);

LAST_ACK – отдалечената страна се е изключила, сокетът е затворен; очаква се потвърждение;

FIN_WAIT_2 – сокетът е затворен; очаква се изключване на отдалечената страна (очаква се флаг FIN);

TIME_WAIT – очакване след затваряне на повторното предаване (две съобщения с флаг FIN) за изключване на отдалечената страна.

tcpdump

tcpdump се използва за прихващане на трафика, получаван и изпращан от конкретен интерфейс на каналния слой (например за мрежова карта от Ethernet стандарт). По подразбиране tcpdump привежда интерфейса в режим promiscuous, за да се прихващат всички пакети, пристигнали на мрежовата карта (не само тези с адрес на дестинацията този на интерфейса). След опциите могат да се задават филтри за пакетите. Основните параметри, по които могат да се отсяват пакетите, са:

- host - име на хоста;
- ip - IP адрес;
- proto - протокол;
- net - адрес на мрежа или подмрежа;
- port - адрес на порт;
- src - параметър, касаещ изпращача;
- dst - параметър, касаещ получателя;

Достъпни са следните протоколи: ether, fddi, tr, wlan, ip, ip6, arp, rarp, decnet, tcp и udp.

От операциите за обединение са достъпни and и or, също така могат да се използват скоби за задаване на приоритет. Като не е задължително да се задава host, в много от случаите е достатъчно src или dst.

Wireshark е отличен анализатор на пакети, но усвояването на tcpdump повече способства за изучаването на TCP/IP.

Най-често извикването на tcpdump изглежда по начина:

tcpdump [-c count] [-s snaplen] [-w file] [expression]

Така във файла file ще бъдат записани няколко (count) от първите прихванати кадри, удовлетворяващи маската expression (например: not port 25). При което, ако дължината на кадъра е била повече от snaplen, тогава той ще се изреже до този размер.

Началото на създадения файл ще съдържа заглавия от библиотеката pcap (24 байта), а след това последователно ще се разположат кадрите, като всеки от тях ще се предшества от временен етикет (8 байта), пълна дължина на кадъра (4 байта) и дължина на прихванатите части от кадъра (4 байта). По този начин началото на първия Ethernet кадър (MAC адреса на получателя) ще се намира отместен с 40 относно началото на файла. Форматът на заглавието на всеки файл и заглавието на кадъра може да се прегледа в заглавния файл pcap.h.

Анализирайки стойностите на 13-тия и 14-тия байт вътре за всеки кадър, може да се провери кой е бил протоколът на по-горния слой по отношение на каналния (<1518 – дъжлина на кадъра, 2048 (0800h) – IPv4, 0806h – ARP, 8035h – RARP и т. н.).

Примери за използване на tcpdump:

а) Да се филтрират всички приети и изпратени пакети с IP адрес 192.168.1.1 в диапазона на портовете от 21 до 23:

tcpdump host 192.168.1.1 and portrange 21-23

б) Да се запише трафика на портове 443 и 80 (https и http) във файл:

tcpdump port 443 or 80 -w http_packets

в) Да се изведе трафика на интерфейс eth0, без да се правят dns резолюции и да се изключат пакети от мрежа 10.0.0.0/8, пакети с порт 22 (ssh) и arp пакети:

tcpdump -i eth0 -n net not 10.0.0.0/8 and port not 22 and not arp

21. Приложен слой

Приложният слой (application layer) осигурява непосредствена поддръжка на приложните процеси и програми за крайния потребител, а също така и управлява взаимодействието на тези програми с различни обекти в мрежата. С други думи приложният слой осигурява интерфейс между приложното програмно осигуряване и системата за комуникация. Той предоставя на приложната програма достъп до различни

мрежови услуги, включително и предаването на файлове и работата на електронната поща.

22. Система за имена на домейни - DNS

DNS (*Domain Name System,* система за символни имена на домейните) е важна за работата на Интернет, понеже за свързването с отдалечената система е необходима информация за неговия IP адрес, а за хората е по-лесно да запомнят буквени (обикновено повече осмислени) адреси, отколкото последователността от цифри на IP адреса. Първоначално преобразуването между адреси на домейни и IP адреси се е реализирало с използването на специален текстов файл hosts, който се е създавал централизирано и автоматично се е разпращал на всяка от машините в своята локална мрежа. С растежа на Интернет е възникнала необходимост от ефективен, автоматизиран механизъм, какъвто е станала системата DNS.

В TCP/IP стека се използва DNS, която има йерархична дървовидна структура, допускаща използване в името на домейна произволен брой съставни части. Съвкупността от имена, за които няколко от старшите съставни части съвпадат, образуват домейн (domain) от имена. Имената на домейните се назначават централизирано, ако мрежата е част от Интернет, иначе - локално.

Съответствието между имената на домейните и IP адресите може да се установява както със средствата на локалния хост (с използването на файла hosts), така и с помощта на централизираната DNS система, основана на разпределена база за съответствие от вида «име на домейн – IP адрес».

Име на домейн е и символното име на компютъра.

DNS функционира по схемата "клиент-сървър". В качеството на клиентска част се използва процедурата за разрешаване на имената - resolver, а в качеството на DNS - сървъра (BIND).

DNS е разпределена система. Всеки DNS сървър съхранява имената на следващия слой от йерархията и освен таблицата на изобразяване на имената съдържа препратки към DNS сървърите на своите поддомейни, което опростява процедурата за търсене.

За ускоряване на търсенето на IP адреси в DNS сървърите се прилага процедура за кеширане на преминалите през тях отговори за определено време, обикновено от няколко часа до няколко дни.

Примери за имена на домейни за организации са:

·com – комерсиални организации;

·edu – образователни организации;

·gov – правителствени организации;

·org – некомерсиални организации;

·net – организации за поддръжка на компютърните мрежи.

DNS притежава следните характеристики:

- *Разпределеност на администрирането*. Отговорността за отделните части от йерархическата структура се поема от различни хора или организации.
- *Разпределено съхраняване на информацията.* На всеки възел от мрежата задължително трябва да се пазят само тези данни, които влизат в неговата зона на отговорност и (възможно) адресите на кореновите DNS сървъри.
- *Кеширане на информацията*. Възелът може да съхранява някакво количество данни, които не са за собствената зона на отговорност. Целта е намаляване на натоварването на мрежата.
- *Йерархична структура*, в която всички възли са обединени в дърво, и всеки възел може или самостоятелно да определя работата на по-долу стоящите възли, или да ги делегира (предава) на други възли.
- *Резервиране.* За съхраняването и обслужването на своите възли (зони) отговарят (обикновено) няколко сървъра, разделени както физически, така и логически, като по този начин те осигуряват съхраняването на данните и отказоустойчивост (продължаване) на работата, даже в случай на отказ на един от възлите.

Основни понятия в DNS

Домейнът (domain — област) е възел в дървото от имена, заедно с всички подчинени му възли (ако има такива), т.е. именован клон или поддърво в дървото от имена. Структурата на името на домейна отразява реда на следване на възлите в йерархията. Името на домейна се чете отляво надясно от младшите домейни към домейните с най-висок ранг (в реда на повишаване на значимостта). Кореновият домейн на цялата система представлява точка ('.'), по-надолу следват домейните от първо ниво (географски или тематични), след това — домейните от второ ниво, трето и т. н. (например за адреса bg.wikipedia.org домейн от първо ниво е org, от второ - wikipedia, от трето - bg). На практика точката в края на името често се пропуска, но тя е важна в случаите за разделянето на относителните домейни и FQDN (Fully Qualifed Domain Name, напълно определено име на домейна).

Поддомейнът (subdomain) е подчинен домейн (например wikipedia.org — поддомейн на домейна org, а bg.wikipedia.org е поддомейн на домейна wikipedia.org). Теоретично такова деление може да достига дълбочина 127 нива, а всеки етикет може да съдържа до 63 символа, като общата дължина заедно с точките не бива да достига 254 символа. Но на практика регистраторите на имена на домейни използват по-строги ограничения.

Ресурсният запис е единица за съхраняване и предаване на информация в DNS. Всеки ресурсен запис има име (т.е. е свързан с определено име на домейн, възел в дървото от имена), тип и поле с данни, форматът и съдържанието на което зависи от типа.

Зоната е част от дървото с имена на домейни (включително и ресурсните записи), разположена като единно цяло на някакъв сървър за имена на домейни (DNS сървър), а по-често е едновременно на няколко сървъра. Целта на отделянето на част от дървото в отделна зона представлява предаване на отговорността за съответния домейн на друго лице или организация. Това се нарича делегиране. Като свързана част от дървото, зоната също представлява дърво. Ако се разглежда пространството от имена на DNS като структура от зони, а не като отделни възли/имена, също се получава дърво, като е оправдано да се говори за родителски и дъщерни зони, за старши и подчинени. На практика повечето зони от 0 и 1 слой ('.', edu, com и др.) се състоят от единствен възел, на който непосредствено се подчиняват дъщерните зони. В големите корпоративни домейни (от 2-ри и по-висок слой) понякога се срещат допълнителни подчинени слоеве без да са отделени в дъщерни зони.

Делегирането е операция за предаване на отговорността за част от дървото с имена на домейни на друго лице или организация. С помощта на делегирането в DNS се осигурява разпределено администриране и съхраняване. Технически делегирането се изразява в отделянето на част от дървото в самостоятелна зона и разполагането на тази зона на отделен DNS сървър, управляван от това лице или организация. При това в родителската зона се включват «свързващи» ресурсни записи (NS и A), съдържащи указатели към DNS сървъра на дъщерната зона, а цялата останала информация, отнасяща се до дъщерната зона, се съхранява вече на DNS сървърите на дъщерната зона.

DNS сървърът е специализирано програмно осигуяване за обслужване на DNS, а също така е и компютър, на който това програмно осигуряване се изпълнява. DNS сървърът може да бъде отговорен за някои зони и/или може да пренасочва запитвания (заявки) на по-висшестоящи сървъри.

DNS клиентът е специализирана библиотека (или програма) за работа с DNS. В редица случаи DNS сървърът играе ролята на DNS клиент.

Авторитетност (authoritative) е признак за разполагането на зона на DNS сървър. Отговорите на DNS сървъра могат да бъдат от два типа: авторитетни (когато сървърът заявява, че сам отговаря за зоната) и неавторитетни (Non-authoritative), когато сървърът обработва заявка и връща отговора на други сървъри. Понякога, вместо да предава заявката по-нататък, DNS сървърът може да върне вече известно му (от предишни заявки) съответствие (режим на кеширане).

DNS заявка (DNS query) е запитване от клиента (или сървъра) към сървъра. Запитването може да бъде рекурсивно или нерекурсивно.

DNS системата съдържа йерархия на DNS сървъри, съответстваща на йерархията на зоните. Всяка зона се поддържа като минимум от един авторитетен DNS сървър, на който е разположена информацията за домейна.

Името на домейна и IP адресът не са тъждествени. Един IP адрес може да има множество имена, което позволява поддържането от един компютър на множество от уеб сайтове (т. нар. виртуален хостинг). Обратното също е вярно. На едно име на домейн може да бъде съпоставено множество от IP адреси, като това позволява създаване на балансирано натоварване.

За повишаване на устойчивостта на системата се използват множество сървъри, съдържащи идентична информация, като в протокола се използват средства, позволяващи поддържане на синхронност на информацията, разположена на различните сървъри. Съществуват 13 коренови сървъри, като техните адреси практически не се променят.

DNS протоколът използва в работата си TCP или UDP, с номер на порт 53 за отговори на заявките. Традиционно заявките и отговорите се изпращат във вид на една UDP дейтаграма. TCP се прилага за AXFR (full zone transfer) заявки. Обичайно този механизъм изпълнява репликация на информация на зоната между сървърите, но той също така може да се използва за злонамерени цели (като получаване на различна информация за осъществяване на спам, разпределени DoS атаки и т.н.).

Когато се обръщаме към сървъра с конкретно запитване (например с използване на домейни само от първо и второ ниво), тогава браузърът, използвайки resolver, изпълнява следния алгоритъм:

1. Търси запис с конкретното запитване във файла hosts, ако не открие, тогава
2. Изпраща запитване към известен DNS кеширащ сървър (като правило, локален), ако на този сървър такъв запис не е открит, тогава
3. DNS кеширащият сървър се обръща към DNS-ROOT сървъра със запитване за адреса на DNS сървъра, отговарящ за домейна от първо ниво. Ако получи адреса, тогава
4. DNS кеширащият сървър се обръща към DNS сървъра, отговарящ за домейна от първо ниво, със запитване за адреса на DNS сървъра, отговарящ за домейна от второ ниво. Ако получи адреса, тогава
5. DNS кеширащият сървър изпраща запитване към DNS сървъра, отговарящ за домейна от второ ниво. Ако получи адреса, тогава
6. DNS кеширащият сървър кешира адреса и го предава на клиента.
7. Клиентът се обръща към откритият IP адрес.

23. Hyper Text Transfer Protocol

HTTP (*HyperText Transfer Protocol* е протокол за предаване на хипертекст) — символно-ориентиран клиент-сървърен протокол на приложния слой без запазване на състоянието, използван от услугата World Wide Web.

Основният обект за работата на HTTP е ресурса, към който сочи URI (*Uniform Resource Identifier* – уникален идентификатор на ресурса) в заявката на клиента. Основните ресурси се съхраняват във файлове на сървъра, но могат да бъдат и други логически (например каталог на сървъра) или абстрактни обекти (като например ISBN). HTTP протоколът позволява да се зададе начина на представяне (кодиране) на един и същ ресурс според различни параметри: mime-тип, език и други. Така клиентът и уеб сървърът могат да си обменят двоични данни, въпреки че този протокол е текстов.

Структура на протокола

Структурата на протокола определя, че всяко HTTP съобщение се състои от три части, които се предават в следващата последователност:

1. Стартов ред (Starting line) — определя типа на съобщението;
2. Заглавия (Headers) — характеризират тялото на съобщението, параметрите на предаването и други сведения;
3. Тяло на съобщението (Message Body) — конкретните данни на съобщението. То задължително трябва да се отделя от заглавието с празен ред.

Стартов ред на HTTP

Стартовият ред е задължителен елемент, понеже определя типа на заявката/отговора. Заглавията и тялото на съобщението могат да отсъстват.

Стартовите редове се различават за заявката и за отговора.

Редът на заявката изглежда по начина:

Метод URI HTTP/Версия на протокола

Пример за заявка:

GET /web-programming/index.html HTTP/1.1

Стартовият ред на отговора на сървъра има следния формат:

HTTP/Версия Код на Състоянието [Пояснение]

Например на предишната заявка на клиента за зададената страница сървърът е отговорил с реда:

HTTP/1.1 200 Ok

Методи на протокола

HTTP методът (HTTP Method) е последователност от някакви символи, освен управляващи и разделители, задаваща основна операция с ресурса. Обикновено методът се представя с кратка дума, записана с главни букви (на английски език). Наименованието на метода е чувствително към регистъра.

Всеки сървър е длъжен да поддържа минимум методите GET и HEAD. Ако сървърът не е разпознал посочения от клиента метод, той трябва да върне статус 501 (Not Implemented). Ако за сървъра методът е известен, но той не е приложим за конкретния ресурс, тогава се връща съобщение с код 405 (Method Not Allowed). И в двата случая сървърът трябва да включи в съобщение с отговор в заглавието Allow и списък от поддържаните методи.

Най-използваните методи са GET и POST.

Методите са:

1. OPTIONS - информация за възможностите (видове заявки, параметри);
2. GET – да се получи ресурс;
3. HEAD - като GET, но без тяло (само заглавия);
4. POST - предава информация на сървъра (изпраща данните от формата);
5. PUT - предава информации на сървъра (изпраща ресурса);
6. DELETE - изтрива ресурса на сървъра;
7. TRACE - «ехо» (връща обратно заявката на клиента);
8. CONNECT - тунелиране на някакъв протокол вътре в HTTP;
9. LINK – установява връзка на зададения ресурс с други;
10. PATCH – аналогично на PUT, но само за фрагмент на ресурса.

Прокси-сървър

Прокси се нарича транзитен сървър, който пренасочва HTTP трафика. Прокси сървърите се използват за ускоряване на изпълнението на заявките чрез кеширане на уеб страниците. В локалната мрежа се използва като междумрежов екран и като средство за управление на HTTP трафик (например за блокирането на достъпа до някои ресурси). В Интернет прокси често пъти се използва за *анонимизация на заявките,* като в този случай уеб сървърът получава IP адреса на прокси сървъра, а не на реалния клиент. В съвременните браузъри може да се зададе списък от прокси сървъри и да се превключва между тях според необходимостта (обикновено такава възможност е достъпна с помощта на разширения или плъгини на браузъра).

Кодове на състоянието

Кодът на състоянието информира клиента за резултатите от изпълнението на заявката и определя неговото последващо поведение. Наборът от кодове на състоянието са стандартизирани и всички те са описани в съответните RFC документи.

Всеки код е представен от цяло тризначно число. Първата цифра задава класа на състоянието, а следващите – поредния номер на състоянието. Обикновено с кода на отговора се извежда и кратко описание на английски език.

Заглавия на HTTP

Заглавието на HTTP (*HTTP Header*) представлява ред от HTTP съобщението, съдържащ разделена от двоеточие двойки от вида «параметър-стойност». Форматът на заглавието съответства на общия формат на заглавията на текстови мрежови съобщения на ARPA (RFC 822). Като правило, и браузърът и уеб сървърът включват в съобщението повече от едно заглавие. Заглавията трябва да се изпратят преди тялото на съобщението и да се отделят от него поне с един празен ред (CRLF).

Наименованието на параметъра трябва да се състои поне от един печатен символ (ASCII кодове от 33 до 126). След наименованието веднага трябва да следва символът двоеточие. Стойността може да съдържа всякакви ASCII символи, освен преход на нов ред (CR, код 10) и възврат на каретката (LF, код 13).

Интервалите (като символ) в началото и в края на стойността се обрязват. Последователността от няколко интервала вътре в стойността може да се възприема като

единствен интервал. Регистърът на символите в наименованието и за стойността нямат значение (ако не е предвидено нещо друго за формата на полето).

Всички HTTP заглавия се разделят на четири основни групи:

1. General Headers (Основни заглавия) — трябва да се включват във всяко съобщение на клиента и на сървъра.
2. Request Headers (Заглавия на заявките) — използват се само в заявките на клиента.
3. Response Headers (Заглавия на отговора) — присъстват само в отговорите на сървъра.
4. Entity Headers (Заглавия на същността) — съпровождат всяка същност на съобщението.

Същности (*entity*, понякога се превежда "обект") това е полезна информация, предавана в заявката или в отговора. Същността се състои от метаинформация (заглавия) и непосредствено съдържание (тяло на съобщението).

Заглавията на същността са обособени в отделен клас, за да няма объркване със заглавията на заявката или със заглавията на отговора при предаването на множественото съдържимо (multipart/*). Заглавията на заявката и на отговора, както и основните заглавия, описват съобщението като цяло и се разполагат само в началния блок на заглавието, а заглавията на същността характеризират съдържимото на всяка част по отделно, като се разполагат непосредствено пред нейното тяло.

Тяло на съобщението

Тялото на HTTP съобщението (*message-body*), ако има такова, се използва за предаване на същността, която е свързана със заявката или с отговора. *Тялото на съобщението* (message-body) се различава от *тялото на същността* (entity-body) само в този случай, когато при предаването се прилага кодиране, посочено в заглавието Transfer-Encoding. В останалите случаи тялото на съобщението е идентично с тялото на същността.

Заглавието Transfer-Encoding трябва да се изпраща за задаване на всяко едно кодиране на предаването, използвано от приложението с цел гарантиране на безопасно и правилно предаване на съобщението. Transfer-Encoding е свойство на съобщението, а не на същността, и то може да бъде добавено или премахнато от всяко едно приложение във веригата от заявки/отговори.

Наличието на тяло на съобщението в заявката се отбелязва с добавяне към заглавията на заявката поле на заглавието Content-Length или Transfer-Encoding. Тялото на съобщението (message-body) може да бъде добавено в заявката само тогава, когато методът на заявката допуска тяло на обекта (entity-body).

Всички отговори съдържат тяло на съобщението, възможно е и с нулева дължина, освен отговорите на заявка с метода HEAD и отговорите с кодове на статуса 1xx (Информационни), 204 (Няма съдържимо, No Content), и 304 (Не модифициран, Not Modified).

24. Simple Mail Transfer Protocol

В качеството на транспортно средство за доставка на съобщения в Интернет се използва SMTP (simple mail transfer protocol, прост протокол за предаване на поща), който е бил стандартизиран в RFC 821.

Изпращане на поща по SMTP протокол

От страната на потребителя обикновено една и съща програма изпълнява ролята на IMAP4 клиент и на SMTP клиент на изпращача. Най-разпространени са Mozilla Thunderbird, Gmail, Microsoft Outlook, Mailbird, eM Client, Blue Mail, K-9 Mail и други. При натискане при тях на бутона "изпращане" става формиране на опашка от съобщения (ако се изпраща не единствено писмо) и се установява двустранен сеанс на общуване с SMTP сървъра на провайдера.

SMTP протоколът прави възможна размяната на страните даже по време на един сеанс. Условно е прието да се счита за клиент тази страна, която започва взаимодействието и иска да изпрати поща, а за сървър – тази, която приема заявките. След като клиентът изпрати на сървъра няколко служебни команди и получи положителни отговори на тях, той изпраща на SMTP сървъра тялото на съобщението. SMTP сървърът получава съобщението, добавя в него допълнителни заглавия, с което на практика той е обработил даденото послание, установява връзка със следващия SMTP сървър по маршрута на следване на писмото. Общуването между всеки два SMTP сървъра става по тази същата схема. Инициират се преговори от клиента, сървърът на тях отговаря, а после получава кореспонденцията и "поставя щампа" в тялото на писмото (в неговата заглавна част). Всичко това силно напомня обичайната (с хартиени писма) поща, където сортирането и изпращането на поща се изпълнява от хора.

Ако на някой етап на предаването SMTP клиентът открие, че е невъзможно да се свърже със следващия сървър, тогава той ще се пробва да изпрати съобщението след някакъв интервал от време – 1 час, 4 часа, ден и т. н., до 4 денонощия в общия случай. При което времевите интервали между опитите, като правило, зависят от настройките на програмата, която препраща пощата. Едновременно такъв сървър трябва да уведоми изпращача на съобщението за невъзможността да достави пощата, като му изпрати стандартно писмо "Failed delivery" (доставката е невъзможна) и там ще разкаже за графика си за следващите опити за предвижване на съобщението. Ако свързващата линия не се възстанови за посочения голям интервал от време (например 4 дни), изпратената информация ще се счита за изгубена.

В момента, в който пощата достигне своя пункт за получаване (SMTP сървъра на адресата на съобщението), тя ще бъде сложена в пощенската кутия на абоната, който ще може, когато поиска, в удобно за него време, да я извлече с POP3 или IMAP протокол, в зависимост от това кой от тях се поддържа от провайдера.

Анализирайки "щамповете" в заглавията на полученото писмо, може да се провери откъде е минало, колко време се е предвижвало, как се е наричала пощенската програма на изпращача и още много друго. Тази информация може да се получи в

"Свойства на писмото", например с MS Outlook, или нещо аналогично в другите пощенски клиенти.

Писмото си има ясно разграничена структура. Тялото на пощенското съобщение се състои от заглавна част и текст плюс влагания. До думата "From:" информацията се внася от междинните SMTP сървъри, всеки от които добавя своите данни в самото начало, като така постоянно се увеличива броя на предаваните по-нататък байтове. Средната заглавна част (от "From:" и до "X-UIDL") почти напълно се формира от пощенския агент на изпращача. Изключение от това е полето "Message-Id:", което се формира първо по маршрута на следване на SMTP сървъра. Някои полета са стандартни и желателни за приложение, като From, To, Subject, Reply-To и т. н. А някои други са допълнителни по отношение на описаните в стандарта текстови съобщения в RFC 822 и не са задължителни. Такива полета започват с "X-" и служат за идентификация на пощенския клиент.

При съставянето на пощенското съобщение, потребителят трябва да попълни полето "To", където записва адреса на получателя (записът "From" автоматично се записва в заглавието на писмото). В Интернет съществуват правила на добрия тон, които препоръчват да се посочва темата на съобщението (полето "Subject").

Ако е необходимо да се изпрати поща до няколко адреса „в явен вид“, тогава е необходимо да се използва полето "CC" (carbon copy) или да се въведат, разделени от запетаи в полето "To". Изразът "в явен вид" означава, че човекът, получил такова писмо и прегледал неговите заглавия, може да определи, на кого още то е било изпратено като копие. Ако възникне необходимост да се достави един и същи текст на няколко човека, но така, че всеки от тях да не знае до кого още е адресирано това писмо, тогава се прилага полето "BCC" (blind carbon copy – сляпо копие). В този случай всички получатели, зададени в полето "BCC", разделени със запетая, ще получат съобщението така, че все едно то е предназначено единствено за тях. Работата за "размножаване" на пощата се изпълнява незабележимо за потребителя на пощенския клиент и от първия по маршрута SMTP агент. Те анализират заглавната част на писмото и ако открият попълнени полетата "CC", "BCC", тогава действат в съответствие със своите алгоритми за обработване на пощенски съобщения.

По вече установеното съединение клиентското програмно осигуряване предава командите на SMTP сървъра, очаквайки незабавно да получи отговорите. В арсенала на SMTP клиента, както и на сървъра, влизат около 10 команди, но ако се ползват само пет от тях, вече може легално да се изпрати пощенско съобщение. Това са: HELO, MAIL, RCPT, DATA, QUIT. Употребата им се предполага именно в тази последователност.

HELO (орязана форма от hello, приветствие) е предназначено за идентификация на изпращача, MAIL задава адреса на изпращача, RCPT (recipient – приемащ) – адреса на получателя. След командата DATA и отговора на нея клиентът изпраща на сървъра тялото на съобщението, което трябва да завършва с ред, съдържащ само една точка.

Сървърът разбира за прекратяването на изпращане на съобщението, когато срещне следващата последователност от символи "\r\n.\r\n" или "<CRLF>.<CRLF>" (CRLF – така наречения възврат на каретката с преход на нов ред, обикновено това се получава при натискане на клавиша Enter за повечето редактори).

За TCP порт номер 25 служи за осигуряване на взаимодействието по SMTP протокол.

Непосредствено след установяване на съединението сървърът извежда ред с код на отговора 220. Като отговор на нея клиентът може да инициира сеанс за връзка по SMTP протокол, като изпрати командата HELO (може и с малки букви) и трябва да посочи в нея като аргумент името на своя компютър. След като приеме командата HELO сървърът е длъжен да направи запитване в DNS и, ако това е възможно, според IP адреса да определи името на домейна на компютъра на клиента (IP адресът става известен в момента на установяването на съединението по TCP протокол).

После в командата "MAIL FROM:" клиентът съобщава обратния адрес на изпращача, който се проверява обикновено само за коректност (това зависи от настройките на SMTP сървъра). След полето "RCPT TO:" следва да се въведе адреса на електронната поща на абоната на дадения сървър. Отговорът "250 <test_1@mail_server.bg> verified" свидетелства за съществуването на съответния логин със зададеното име. Клиентът изпраща командата DATA и очаква покана, за да започне да прехвърля тялото на писмото (код 354).

Съобщението може да бъде дълго, но задължително трябва да свършва с ред, в който има една единствена точка. Последното служи за сигнал на SMTP сървъра за това, че тялото на писмото е приключило. Той присвоява на писмото определен идентификатор и чака командата QUIT, след което сеансът се счита за приключил.

Как ще действат потребителското и сървърното програмно осигуряване в случай на необходимост от "масово" разпращане? Ако клиентът изпраща съобщение, в което в заглавната част от полето CC са посочени няколко e-mail адреса, първият по маршрута на предаване SMTP сървър трябва да установи сеанс за предвижването на пощата до всеки един от сървърите от дадения списък и да изпрати точно копие на писмото до всеки. В случай на използване на полето BCC клиентът, формиращ съобщението, ще унищожи записа BCC в тялото на съобщението и няколко пъти (според броя на адресите) ще изпрати на първия SMTP сървър командата "RCPT TO:" всеки път с нов адрес в качеството на аргумент. Така сървърът ще получи указание да разпрати пощата на множество от адреси. В този случай получателите на писмата нищо няма да знаят един за друг, понеже разпращането се е реализирало посредством командите на SMTP протокол (без използване на информацията в заглавната част на писмото).

Протоколът SMTP съществува от 1982 година (RFC 821). Впоследствие в него са били внесени някои допълнения, изразили се в появата на ESMTP (Extended SMTP, RFC 1425 през 1993 г.). Ако клиентът поддържа ESMTP, тогава първата му команда ще бъде EHLO (Extended Hello), а не HELO. Като отговор на нея сървърът е длъжен да изведе списък на всички допълнителни команди, които той умее да обработва. Клиентът може или да се възползва от тях, или да ги игнорира, като изпрати съобщение за това.

25. Мрежи на бъдещето

Програмното конфигуриране на мрежите (Software-Defined Networking, SDN) и виртуализацията на мрежовите функции (Network Functions Virtualization, NFV) са едни от най-обсъжданите теми през последните години в бранша на компютърните мрежи.

Как се развиват тези идеи във времето?

- През лятото на 2011 г. проведените изследвания от мрежовите оператори потвърждават, че виртуализацията на мрежите осигурява необходимата производителност за поддръжката на реални натоварвания;
- През месец април на 2012 г. мрежовите оператори обсъждат въпросите за сътрудничество в NFV направлението на конференция в Сан Франциско Open Network Sammit;
- През юни на 2012 г. на среща в Париж мрежовите оператори стигат до споразумението за организацията на нов отраслов форум;
- През септември на 2012 г. под егидата на Европейския институт за телекомуникационни стандарти (European Telecommunications Standards Institute, ETSI) е създаден NFV форум;
- През октомври на 2012 г. е публикуван първият официален документ;
- През месец януари на 2013 г. ETSI провежда първата пленарна сесия на ETSI NFV Industry Specification Group;
- През октомври на 2013 г. с участието на 25 оператора са подготвени първите NFV ISG документи и вторият официален документ за NFV;
- През септември на 2014 г. стартира отворената платформа Open Platform for NFV Project (OPNFV);
- През месец януари на 2015 г. излиза второто издание на NFV ISG документите.

В NFV виртуализацията се прилага към функционалността на мрежите. Според концепцията за виртуализация на мрежовите функции, частното мрежово оборудване и специализираните мрежови устройства се заменят с програмно осигуряване, което се изпълнява на сървъри с общо предназначение, т.е. на комерсиални стандартни платформи.

Този път инициаторите за стандартизация са мрежовите оператори. Това са големите телекомуникационни компании.

NFV е идея на мрежовите оператори, която цели отказ от употреба на конкретно производствено мрежово оборудване, с цел да отпадне зависимостта от доставчика. В традиционните мрежи се налага установяване на фрагментирано специализирано оборудване. Необходимостта от скъпи разработки на нови апаратни решения затруднява излизането на пазара на нови играчи, възпрепятства иновациите и конкуренцията. Като NFV предполага използване на стандартни сървъри и суичове, виртуални устройства, независими от доставчиците и програмни инструменти за управление и оркестрация.

Според ETSI вече се транформира начинът, по който мрежовите оператори изграждат и експлоатират мрежите и мрежовите услуги, като прилагат стандартни технологии за виртуализация. Също така се консолидират различни видове мрежово оборудване върху стандартни и мощни сървъри, комутатори и устройства за съхранение. NFV трансформира мрежовите архитектури, реализирайки мрежовите функции в

софтуера, работещ на стандартни сървъри. Този софтуер може динамично да се премества или да се репликира в различни примери на различни места в мрежата, без да се налага инсталиране на ново оборудване. NFV допълва SDN в управлението на мрежите. Те разчитат на различни методи: докато SDN разделя контрола от пренасочването на данните, предлагайки централизиран изглед върху мрежата; NFV основно се фокусира върху оптимизиране на самите мрежови услуги.

Open Platform for the NFV Project (OPNFV) е еталонна платформа с отворен код за NFV. Управлява се от Linux Foundation и се поддържа от AT&T, China Mobile, Cisco, Ericsson и др. Проектът OpenFlow е в основата на SDN. Потребителите дефинират потоците данни и пътищата им, независимо от инфраструктурата под тях (маршрутизатори и комутатори). Това е проект с отворен код в сътрудничество между Stanford University и University of California at Berkeley.

Виртуалната среда за мрежова инфраструктура е елемент от OSS/BSS (Operations Support Systems/Business Support Systems) решенията и проектите. OSS/BSS системите са предназначени за комплексно управление на телекомуникационните ресурси на компанията.

С развитито на отрасъла на мрежовите услуги решаващ фактор в конкурентната борба между мрежовите оператори стават услугите, които те могат да предоставят. Поради тази причина оперативността и качеството на услугите получават ново значение. В резултат функционалносттта на системите за експлоатационна поддръжка на телекомуникационните мрежи значително се разширява и се появява нов клас ИТ решения - OSS/BSS системите.

Съвременните OSS/BSS системи съдържат множество модули (класове) и подсистеми, насочени към решаването на различни бизнес задачи. Комбинирането на различните класове с корпоративните информационни системи (CRM, HelpDesk и др.) осигурява необходимата функционалност за решаването на конкретните задачи.

Въпроси, задачи и проблеми за решаване

NFV и SDN представляват крачка напред в направлението за виртуализация на компютърните мрежи и приложения. Преходът към тези технологии ще позволи да се внедрят множество нови потребителски услуги и да се намалят както операционните, така и капиталовите разходи. Телекомуникационният отрасъл ще получи предимствата и гъвкавостта на облачните изчисления. Използването на отворена и свободна платформа, поддържаща интеграция с различни Open Source продукти, ще позволи да бъдат удовлетворени разнообразните нужди на разработчиците на ИТ решения.

OPNFV трябва да предостави интегрируема, свободна и еталонна платформа за операторския слой, над който ще работят всички участници в отрасъла, като целта е да се продължи еволюцията на NFV и да се осигури еднородност, производителност и съвместимост с множество други свободни компоненти. Предполага се, че OPNFV ще увеличи производителността и ще намали разходите на енергия, ще подобри надеждността, достъпността и удобството при обслужването и предоставянето на инструментариум с широк обхват на възможностите. Понеже част от свободните

компоненти, с които се планира интеграцията вече е създадена, затова първоначално OPNFV ще е насочена към създаването на инфраструктура (NFV Infrastructure, NFVI) и системи за управление на виртуализираната инфраструктура (Virtualized Infrastructure Management, VIM), използвайки тези компоненти.

Кои са направленията за развитие на OSS/BSS системите, целящи осигуряване на виртуализация на мрежовата инфраструктура?

Това са:

- Стремеж към отворена и модулна архитектура;
- Ефективно построяване на бизнес процесите и безопасно управление на информационните потоци;
- Гъвкава платформа и поддръжка на решения на различни производители;
- Мащабируемост на системата и възможност за разширяване;
- Еволюция на OSS/BSS системите в посока на клиентно ориентирания подход.

Към потенциалните предимства на NFV могат да се изтъкнат следните възможности:

- Бързо разгръщане, модернизация и изключване на мрежовите функции;
- Гъвкаво комбиниране на няколко мрежови функции на една сървърна платформа (което е особено важно в многоарендна среда);
- Разполагане на мрежовите функции там, където това е по-ефективно или по-икономично;
- Съкръщаване на разходите благодарение на използването на стандартни сървъри вместо на частно мрежово оборудване.

Освен това NFV осигурява бързо мащабиране на услугите, възможност за използване на програмно осигуряване за получаване на печалба, използване на евтини стандартни сървъри вместо на мрежово оборудване, повишаване на операционната ефективност, оптимизация на мрежата в реално време, икономия на електроенергия за сметка на консолидацията на натоварването. Новите мрежови функции могат да се разработват от малки компании, а това ще способства за иновациите.

Моделът на OPNFV архитектурата засега само се обсъжда. Отрасловата организация на OPNFV най-вече цели да взаимодейства с ETSI и да използва Open Source елементи. От самото начало за OPNFV основен интерес са представлявали апаратните платформи (изчислителни ресурси, системи за съхраняване на данни и мрежово оборудване), слоят за виртуализация, виртуалните изчисления, компютърните мрежи и ресурсите за съхраняване на данни, а също така и средствата за управлението на виртуализираната инфраструктура. Обаче много по-важно значение могат да имат оркестрацията и управлението на по-високо ниво (Management and Orchestration, MANO). Повечето мрежови оператори не бързат с внедряването на NFV, защото редица въпроси, свързани с MANO, остават отворени.

Експертите считат, че все още не е ясно, каква оркестрация ще се изисква за VNF в мрежите и как ще се разделят функциите за управление между различните нива на архитектурата на NFV.

Класическият мрежов модел предполага използването на три плоскости: данни, контрол и управление. Цялата идея на SDN се състоит в това, че трябва да се променят пропорциите между двете последни плоскости за управление на по-ниско и по-високо ниво. Все повече функции ще преминават на нивото за контрол и ще стават програмируеми и автоматизируеми. Целта е не конфигуриране или програмиране на компютърните мрежи, а програмиране на услуги с използването на автоматизацията.

При това съществуват два основни подхода. Според първия, протоколите от плоскостта за контрол се заменят на API в централен контролер на SDN, а механизмите за препращане (forwarding) в плоскостта на данните се свеждат до прости операции в SDN комутатора. Това предполага замяна на съществуващото оборудване. При втория подход SDN допълва съществуващите технологии, като за оптимизация на производителността се въвеждат допълнителни операции за управление, а за създаването на нови услуги се използват програмни средства, в резултат съществуващата мрежа става по-програмируема.

Повечето мрежови оператори и производители на мрежово оборудване поддържат първия подход, а вторият се прилага от изследователските организации.

NFV според същността си е нова методология, нов подход, при което е не само технологичен, но и организационен. Този подход предполага не просто замяна на съществуващото мрежово оборудване на стандартни сървъри, а преход към изключително сложни процедури и процеси. За да се вземе решението за внедряването на тази методология, мрежовият оператор трябва по нов начин да изгради своята дейност. Преходът от лабораторното тестване към реалното използване и получаването на печалба от внедряването става нетривиална задача. При миграцията на прехода от съществуващата мрежова инфраструктура към NFV може да представлява значителна трудност, както и изменението на системите за управление на OSS/BSS.

Общ тренд в отрасъла представлява направлението за виртуализация на мрежите и използването на новите концепции, залагащи на програмното управление. Лидиращата концепция е създаването на нова мрежа с програмно-конфигурируеми SDN мрежи. Тук се подчертава връзката на двата подхода за мрежова виртуализация NFV и за програмно управление на SDN мрежата.

Най-важната разлика между NFV и SDN е крайната цел на концепцията. Ако при NFV се планира конкретните мрежови функции да бъдат реализирани програмно, като след това те да се управляват вече като програмни обекти, то SDN е идеология за работа на цялата компютърна мрежа, където управлението и отговорността за вземане на решенията (за маршрутизация, комутация и т.н.) са изнесени на отделен централизиран слой. С други думи NFV са конкретни програмни компоненти, реализиращи конкретни мрежови функции, а SDN е идеология за работата на цялата компютърна мрежа и за взаимодействието на нейните функционални слоеве.

Важното е, че NFV и SDN могат да се внедряват както независимо една от друга, така и съвместно, като се допълват взаимно. Също така за NFV и SDN това, по което те много си приличат, е в революционното си влияние на изменението на принципите за управление на информационните комуникации и основите за построяване на BSS/OSS за най-новите компютърни мрежи.

Използвани източници

RFCs, https://www.ietf.org/standards/rfcs/

1. RFC 768, User Datagram Protocol, https://www.rfc-editor.org/rfc/pdfrfc/rfc768.txt.pdf
2. RFC 791, Internet Protocol, https://www.rfc-editor.org/rfc/pdfrfc/rfc791.txt.pdf
3. RFC 792, Internet Control Message Protocol, https://www.rfc-editor.org/rfc/pdfrfc/rfc792.txt.pdf
4. RFC 793, Transmission Control Protocol, https://www.rfc-editor.org/rfc/pdfrfc/rfc793.txt.pdf
5. RFC 826, An Ethernet Address Resolution Protocol, https://www.rfc-editor.org/rfc/pdfrfc/rfc826.txt.pdf
6. RFC 894, A Standard for the Transmission of IP Datagrams over Ethernet Networks, https://www.rfc-editor.org/rfc/pdfrfc/rfc894.txt.pdf
7. RFC 1034, Domain names - concepts and facilities, https://www.rfc-editor.org/rfc/pdfrfc/rfc1034.txt.pdf
8. RFC 1035, Domain names - implementation and specification, https://www.rfc-editor.org/rfc/pdfrfc/rfc1035.txt.pdf
9. RFC 7323, TCP Extensions for High Performance, https://www.rfc-editor.org/rfc/pdfrfc/rfc7323.txt.pdf
10. RFC 2328, Open Shortest Path First, https://www.rfc-editor.org/rfc/pdfrfc/rfc2328.txt.pdf
11. RFC 4271, Border Gateway Protocol, https://www.rfc-editor.org/rfc/pdfrfc/rfc4271.txt.pdf
12. RFC 4291, IP Version 6 Addressing Architecture, https://www.rfc-editor.org/rfc/pdfrfc/rfc4291.txt.pdf
13. RFC 5321, Simple Mail Transfer Protocol, https://www.rfc-editor.org/rfc/pdfrfc/rfc5321.txt.pdf
14. RFC 7231, Hypertext Transfer Protocol, https://www.rfc-editor.org/rfc/pdfrfc/rfc7231.txt.pdf

Wikipedia, https://en.wikipedia.org/

15. Border Gateway Protocol, https://en.wikipedia.org/wiki/Border_Gateway_Protocol
16. Data link layer, https://en.wikipedia.org/wiki/Data_link_layer
17. Domain Name System, https://en.wikipedia.org/wiki/Domain_Name_System
18. Ethernet, https://en.wikipedia.org/wiki/Ethernet
19. Hypertext Transfer Protocol, https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol
20. Internet Control Message Protocol, https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol
21. Internet protocol suite, https://en.wikipedia.org/wiki/Internet_protocol_suite
22. IP address, https://en.wikipedia.org/wiki/IP_address
23. IPv4, https://en.wikipedia.org/wiki/IPv4
24. IPv6, https://en.wikipedia.org/wiki/IPv6
25. Open Shortest Path First, https://en.wikipedia.org/wiki/Open_Shortest_Path_First
26. OSI model, https://en.wikipedia.org/wiki/OSI_model
27. Routing Information Protocol, https://en.wikipedia.org/wiki/Routing_Information_Protocol
28. Routing, https://en.wikipedia.org/wiki/Routing
29. Simple Mail Transfer Protocol, https://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol
30. Transmission Control Protocol, https://en.wikipedia.org/wiki/Transmission_Control_Protocol
31. User Datagram Protocol, https://en.wikipedia.org/wiki/User_Datagram_Protocol

Cisco, https://www.cisco.com/

32. Easy network, https://easy-network.ru/
33. Welcome to the Cisco Learning Network, https://learningnetwork.cisco.com/welcome/

СЪДЪРЖАНИЕ