Факултет инжењерских наука Универзитета у Крагујевцу

Назив студијског програма: Машинско инжењерство

Ниво студија: Основне академске студије

Модул: Информатика у инжењерству

Предмет: Архитектура рачунарских система

Број индекса: 711/2012

Саша В. Ђорђевић

Основни принципи мрежних протокола

завршни рад

Комисија за преглед и одбрану:

1. **Др Јасна Радуловић - ментор**
2. _____
3. _____

Датум одбране:_____

Оцена:_____

У оквиру овог завршног рада кандидат треба да:

Истакне улогу протокола и слојева у архитектури рачунарских мрежа, начин рада OSI референтног модела и TCP/IP референтног модела.

Такође, потребно је описати циљеве у пројектовању слојева. На крају треба анализирати и мрежну опрему која је такође важан сегмент у преносу поруке, као што су *Router, Hub u Switch.*

Препоручена литература:

[1] **Стојановић Д.** (2005) Увод у рачунарство, Универзитет у Нишу, Електронски факултет, Ниш.

[2] **William Stallings.** (1996) Computer Networking with Internet Protocols and Technology.

[3] **Douglas E. Comer, David L. Stevens.** (2000) Internetworking with TCP/IP, Client-Server Programming and Applications.

Крагујевац, ментор:

04.09.2013. **Др Јасна Радуловић**

Резиме: У раду се описују „Мрежни протоколи“. Кроз овај рад је детаљно објашњено шта су то уствари мрежни протоколи, чему служе, које врсте протокола постоје и који су то слојеви на којима ради OSI референтни модел. Постоји велики број протокола од којих сваки има своју посебну улогу.

Кључне речи: Протокол, *ОСИ* референтни модел, *ФТП* - фајл трансвер протокол, *ДНС* – Домен нејм сервис, *ТЦП/ИП* – Трансвер контрол протокол / Интернет протокол, *Telnet, , НетБИОС, Рутер, Хуб, Свич*.

Abstract: We describe the "Network Protocols". Through this work is explained in detail what they actually network protocols, which are used, which are types of protocols and what are the layers of employing OSI reference model. There are a number of protocols, each of which has its special role.

Key words: Protokol, *OSI* reference model, *FTP* – File Transfer Protocol, *DNS* – Domain Name Service, *TCP/IP* – Transfer Control Protocol/Internet Protocol, Telnet, *NetBIOS*, Router, Hub, Switch.

САДРЖАЈ

1. УВОД

1.1. Архитектура протокола

Концепти дистрибуираног процесирања и рачунарских мрежа условљавају да целине у различитим системима имају потребу да међусобно комуницирају. Примери целине су: кориснички апликациони програм, пренос датотека, системи за управљање базама података, уређаји за електронску пошту и терминали. Пример система су: рачунар, терминал и удаљени сензор. У неком случају целина и систем су коегзистентни. У општем случају целина је било шта способно да шаље и прима информације, а систем је физички одвојен објекат који садржи једну или више целина.

Две целине које треба да комуницирају „морају да говоре истим језиком". Шта комуницира, како се комуницира и када се комуницира мора да се повинује неком опште прихватљивом скупу конвенција целина, које су укључене у комуникацију. Скуп конвенција назива се протокол, и он се може дефинисати као скуп правила која управљају разменом података између две целине. Кључни елементи протокола су:

- синтакса: укључује формат података, кодирање и нивое сигнала;
- семантика: укључује управљачке информације за координацију и вођење рачуна о грешкама;
- временска синхронизација: укључује подешавање брзина и секвенционирање.

Подаци које треба разменити морају се слати у рамовима специфичног формата (синтакса). Управљачко поље обезбеђује различите регулаторске функције као што су постављање врсте рада и успостављање везе (семантика).

Резултати стандарда су следећи:

- произвођачи се подстичу да примене стандарде због очекивања да ће се, због широке употребе стандарда, њихови производи више продавати,
- корисници захтевају да стандарде примењују сви произвођачи јер могу опрему набављати од било кога од њих.

Задатак да се обезбеди комуникација између апликација на различитим рачунарима превише је комплексан да би се посматрао као једна целина. Проблем се мора раздвојити у целине које се могу лакше реализовати. Тако, пре него што неко развије стандарде, мора да постоји структура или архитектура која дефинише комуникационе циљеве [1].

Интернационална организација за стандардизацију (*ISO*) је 1977. године формирала комисију са задатком да развије такву архитектуру. Резултат је отворени систем за међусобно повезивање (*OSI*), референтни модел који је оквир за дефинисање стандарда за повезивање хетерогених рачунара.

OSI модел обезбеђује основу за повезивање отворених система дистрибуираних апликативних процеса. Термин „отворен" одређује способност да било која два система која подржавају референтни модел и одговарајуће стандарде могу међусобно да се повежу.

1.2. Мрежна архитектура

Због комплексности пројектовању многе мреже су организоване као низ слојева или нивоа. Број слојева, као и њихова имена разликују се од мреже до мреже. У свим мрежама намена сваког слоја је да пружи одређену услугу вишим слојевима, штитећи те нивое од детаља (нпр. како се услуга реализује).

Слој N на једном рачунару води конверзацију са слојем N на другом рачунару. Правила и конвенције које се користе у овој конверзацији заједничке су и познате као протокол N-тог слоја.

Практично, подаци се не шаљу директно из слоја N једног рачунара ка слоју N другог рачунара (осим најнижи слој). Уместо тога, сваки слој шаље податке и управљачке информације ка слоју који је одмах испод, докле год не стигне до најнижег слоја. На најнижем слоју постоји физичка комуникација са другим рачунаром, супротно од виртуелне комуникације коју користе виши слојеви. Између сваког пара суседних слојева постоји интерфејс. Интерфејс слојева дефинише које основне операције и услуге нижи слој нуди вишем слоју N. Када пројектант мреже одлучи колико слојева ће укључити у мрежу и шта ће који од њих да ради, најважније је јасно дефинисати интерфејсе између слојева. Јасно дефинисање слојева, заузврат, захтева да сваки слој извршава специфични скуп јасно дефинисаних функција. Последица тога је да се може један слој заменити потпуно другим слојем (нпр. телефонска линија се може заменити сателитском везом), пошто је све што се тражи од новог слоја да пружи исти скуп услуга за слој изнад себе као што је претходно био обезбеђен [2].

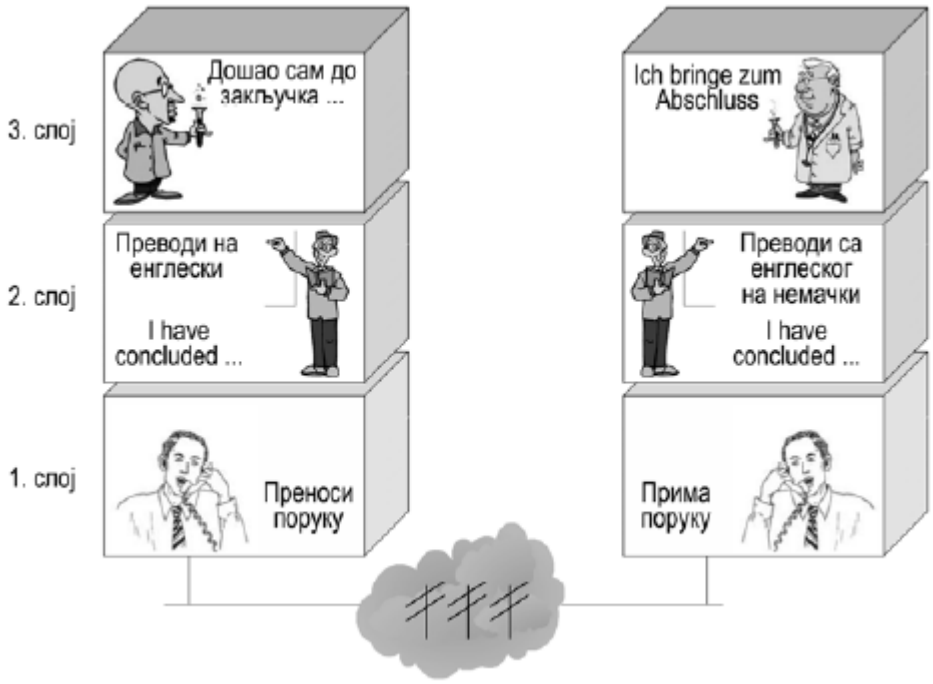
Скуп слојева и протокола назива се слојевита архитектура.

Спецификација архитектуре мора да садржи довољно информација да омогући ономе ко њу имплементира да изгради програм за сваки слој, тако да програм исправно реализује одговарајући протокол. Ни детаљи примене, ни спецификација интерфејса нису делови архитектуре. У ствари, чак ни сви интерфејси на свим рачунарима нису исти, а ипак је обезбеђено да сваки рачунар може правилно да користи протокол. Пример вишеслојне архитектуре је да два истраживача (парњаци слоја) од којих је један у Београду а други у Минхену, желе да комуницирају (слика 1.). Пошто немају заједнички језик они укључују преводиоце (парњак - процес 2. слоја), а сваки од њих контактира инжењере (парњак- процес 1. слоја).

Истраживач 1 жели да пренесе своје закључке свом парњаку. Он шаље своје закључке на српском језику као поруке преко интерфејса 3. и 2. слоја, ка свом преводиоцу који реченицу „Дошао сам до закључка...” преводи у „i have concluded.... ”, или на француски језик, зависно од протокола 2. слоја.

Преводилац затим даје поруку свом техничару који поруку преноси телеграмом, телефоном, рачунарском мрежом или неким другим средством, зависно како су се унапред договорили (протокол 1. слоја).

Када порука стигне она се преводи на немачки и преноси преко интерфејса 2. и 3. слоја истраживачу 2. Уочимо да је сваки протокол комплетно независан од других докле год се интерфејс не мења. Преводилац се може мењати од енглеског на француски подразумевајући да се оба слажу и да ни један не мења интерфејс ка слоју 1 или 3.

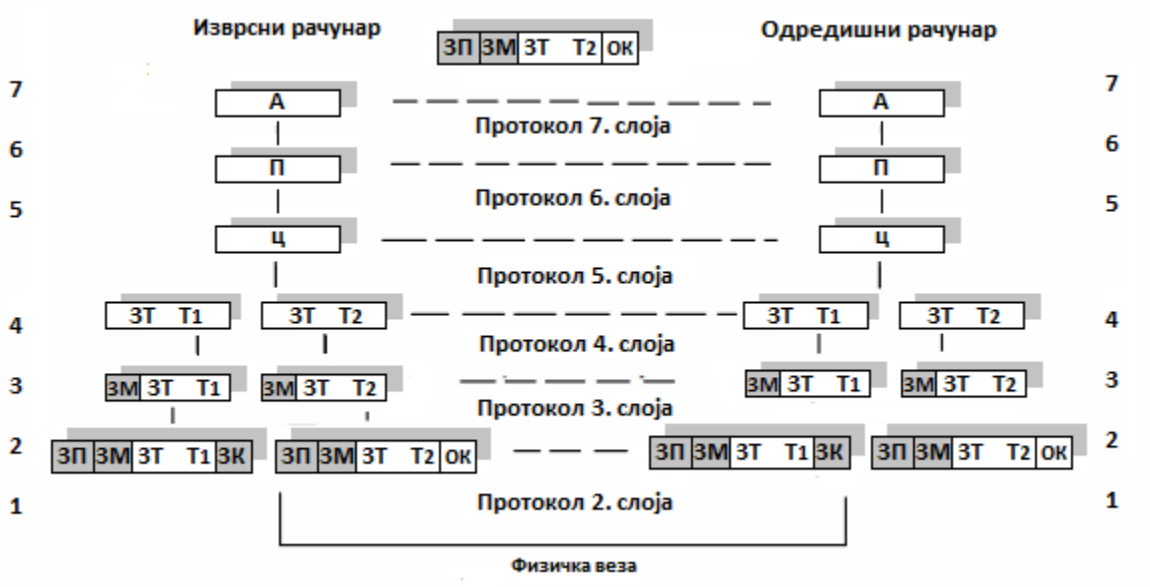


Слика 1. *Пример комуникације истраживача – парњака.*

Посматрајмо технички пример: како обезбедити виртуелну везу ка највишем слоју седмослојне мреже. На слици 2 поруку А генерисао је процес који се одвија на 7. слоју. Порука се заједно са заглављем прослеђује са 7. до 6. слоја преко интерфејса 6/7 слоја.

У овом примеру 6. слој трансформише поруку, на пример врши компресију текста, а затим прослеђује нову поруку П ка 5. слоју преко интерфејса 5/6 слоја. Слој 5, у овом примеру, не модификује поруку, већ једноставно регулише смер преноса (нпр. спречава долазеће поруке да дођу у 6. слој док је он заузет и води рачуна о серији одлазећих порука ка 5. слоју.

У овом примеру, као и у другим стварним мрежама, не постоји ограничење у величини поруке коју може 4. слој да прими, али постоји ограничење које поставља 3. слој. Посматрајући даље, 4. слој мора да разбије долазећу поруку Т у мање јединице, додајући заглавље (ЗТ) свакој јединици. Ово заглавље укључује управљачке информације, као што је нпр. редни број1, да би омогућио 4. слоју на одредишном рачунару да повеже делове у правилном редоследу.



Слика 2. *Пример виртуелне везе.*

На 3. слоју свакој поруци се додаје заглавље (ЗМ) и прослеђује 2. слоју. Други слој поред заглавља (ЗП) додаје и ознаку о крају поруке (ОК). Тако формиран рам прослеђује се физичком везом.

Важна ствар за разумевање слике 2 је веза између виртуелне и стварне комуникације и разлика између протокола и интерфејса. Процес парњак 2 на 7. слоју, на пример, размишља о својој комуникацији као да је „хоризонтална", користећи протокол 7. слоја. Сваки од њих има процедуру која се назива „пошаљи на другу страну" и процедуру „узми са друге стране", иако ове процедуре стварно комуницирају са нижим слојевима преко интерфејса 7/6 слоја, а не са другом страном.

Апстракција са процесима парњака је суштинска за пројектовање мрежа. Без ове технике не би било могуће раздвојити пројектовање комплетне мреже (нерешиви проблем) у неколико мањих целина, које се могу надгледати, пројектовати и које се називају пројектовање појединих слојева.

2. Циљеви у пројектовању слојева

Сваки слој мора да има механизам за успостављање везе. Пошто се мрежа нормално састоји од много рачунара, а неки од њих садрже више процеса, потребан је начин да се процесу на једном рачунару укаже са ким он то жели да комуницира. Адресирање је потребно у било ком слоју где постоји више одредишта.

Као што постоји механизам за успостављање везе кроз мрежу, тако постоји и механизам за раскидање везе, када она више није потребна.

Други скуп одлука у пројектовању су правила за пренос података. Подаци могу да се преносе:

- само у једном смеру, што се назива симплекс (једносмерна) комуникација,
- у оба смера (али не истовремено), што се назива полудуплекс комуникација, или
- у оба смера истовремено, што се назива потпуна дуплекс (истовремена двосмерна) комуникација.

Протокол мора, такође, да одреди колико логичких канала вези припада и какви су приоритети. Многе мреже обезбеђују најмање два логичка канала по једној вези, на пример: један за нормалне податке а један за брзе податке, један за корисничке а један за управљачке податке, један за један а други за други смер.

Контрола грешака важна је компонента када комуникациони путеви нису савршени. Познати су многи кодови за детекцију и корекцију грешака, али оба краја везе морају се договорити који се користи. Такође, прималац мора да има начин да укаже пошиљаоцу које поруке су коректно примљене а које нису.

Није тачно да баш сви комуникациони канали задржавају редослед порука које се шаљу. Да би се могао разрешити могући губитак редоследа, протокол мора да направи експлицитну припрему пријемнику да би омогућио да се делови могу исправно повезати. Видљиво решење је пребројавање делова, али ово решење још увек оставља отвореним питање шта треба урадити са деловима који стижу ван редоследа. Како спречити брзог пошиљаоца да преплави подацима спорог примаоца. Различита решења постоје. Сва укључују неку врсту повратне спреге од пријемника ка предајнику, директно или индиректно, зависи од тога каква је текућа ситуација пријемника.

Други проблем који се мора разрешити на различитим нивоима је способност процеса да прихвати произвољно дугачке поруке. Ова карактеристика доводи нас до механизма разлагања (сегментација), слања и поновног обнављања (асемблирање) поруке. Проблем који се јавља је и шта радити када процес инсистира на слању јединица података које су тако мале да је слање сваке понаособ неефикасно.

Овде је решење скупити више малих порука које се шаљу ка заједничком одредишту у једну велику поруку, и раздвојити велику поруку на другом крају.

Када није згодно, или је скупо успостављање одвојених путева за сваки пар комуникационог процеса, слој испод може да одлучи да користи исти пут за више различитих веза. Ово мултиплексирање и демултиплексирање, уколико се ради транспарентно, може да користи било који слој. Мултиплексирање је потребно 1. слоју, на пример, где се сав саобраћај за све везе мора послати преко једног, или у најбољем случају неколико физичких канала.

Када постоји неколико могућих путева између изворишта и одредишта на истој тачки хијерархије, одлука о рутирању мора се донети. На пример, при слању података из Лондона у Москву одлучивање на вишем нивоу упутиће их преко Француске или Немачке, у зависности њихових закона, а одлучивање на нижем нивоу помоћи ће да се одабере расположиви канал у зависности од текућег саобраћаја.

3. Интерфејс и услуга

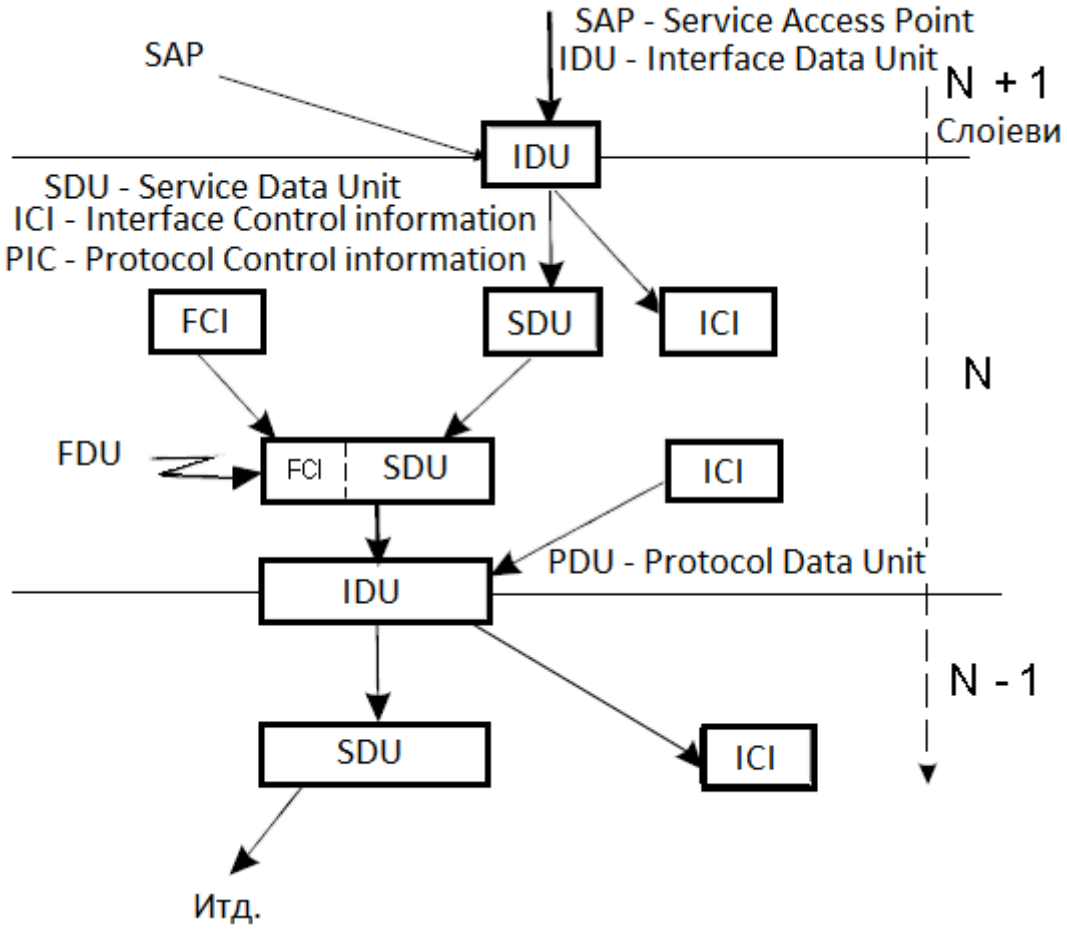
Функција сваког слоја је да обезбеди услугу слоју изнад себе. Активни елеменат у сваком слоју обично се назива целина. Целина може бити софтвер (нпр. процес) или хардвер (као што је интелигентни улазно / излазни чип). Целине на истом слоју, на различитим машинама називају се парњак целине. Целина у слоју N имплементира услугу који користи слој N+1. У овом случају слој N назива се давалац услуге, а слој N+1 назива се корисник услуге. Слој N може да користи слој N-1 да би обезбедио себи услугу. Могуће је обезбедити више класа услуга нпр. брза, скупа комуникација, или спора, јефтина.

Услуга се добија на тачкама приступа услузи које се означавају са *SAP*. Тачка приступа услузи N-тог слоја је место где слој N+1 може да приступи понуђеној услузи. Свака тачка приступа има адресу која је једнозначно идентификује. Тачке приступа у телефонском систему су утичнице у које се телефони могу прикључити, а адреса тачке приступа је телефонски број утичнице. Да бисте некога позвали морате знати његову адресу тачке приступа. У поштанском систему, адреса тачке приступа је улична адреса или поштански фах. Када треба послати писмо мора се знати адресу онога коме је писмо упућено [3].

Да би два слоја размењивала информације мора да постоји усаглашени договор око скупа правила везаних за интерфејс. Интерфејс целине слоја N+1 преноси јединице података интерфејса (које означавамо као *IDU*) ка N-том слоју преко тачка приступа (*SAP*), као што је приказано на слици 1.3. Јединица података интерфејса (*IDU*) састоји се од јединице података услуге (*SDU*) и управљачких информација. Јединица података услуге је информација која се прослеђује преко мреже ка парњак целини и затим ка N+1 слоју.

Управљачке информације потребне су нижим слојевима да би обавиле функцију као што је нумерисање бајтова у јединица података услуге али нису део самих података. Да би пренела јединицу података услуге, целина слоја може да је дели у неколико делова. Сваки од тих делова може да има своје заглавље и да буде послат као посебна јединица података протокола (*PDU*).

Заглавље јединица података протокола користе парњак целине да би реализовале свој парњак протокол. Он идентификују која јединица протокола садржи податке, која садржи управљачке информације, која садржи редне бројеве, итд.



Слика 3. *Пренос јединица података између слојева.*

3.1. Услуге са и без успоставе везе

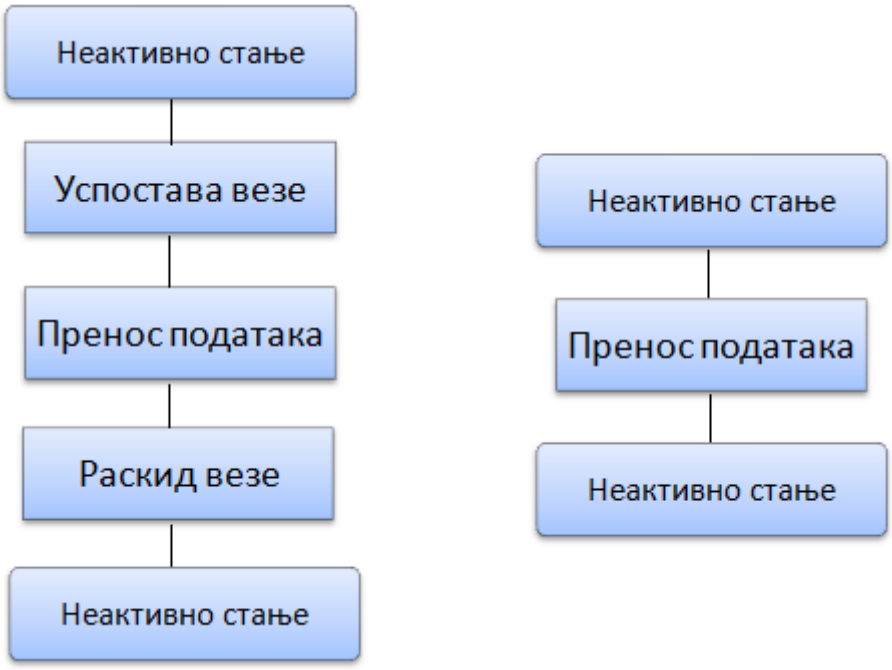
Слојеви нуде два различита типа услуга ка слоју изнад њих: са успоставом везе и без успоставе везе. Услуга са успоставом везе је моделиран по узору на телефонски систем у коме да бисте са неким разговарали подижете слушалицу, окрећете број, разговарате и на крају разговора спуштате слушалицу. Слична је употреба услуге са успоставом везе: корисник услуге прво успоставља везу, користи везу и раскида је.

Суштински веза функционише као цев: пошиљалац шаље објекте (битове) на једном крају, а пријемник их преузима у истом редоследу на другом крају. Услуга без успоставе везе моделирана је по узору на поштански услугу. Свака порука (писмо) садржи комплетну адресу и свака је упућена (рутирана) независно од осталих. Нормално, када се две поруке шаљу на исто одредиште, прва која се пошаље прва ће и стићи. Може се десити да прва порука која се пошаље закасни тако да друга стигне пре ње. Са системом са успоставом везе ово не може да се деси.

Свака услуга може се окарактерисати преко квалитета услуге (QoS). Неке услуге су поуздане у смислу да је вероватноћа губитка података занемарљива. Обично се, за поуздану услугу користи потврда, тако да је пошиљалац сигуран да је порука стигла на одредиште. Процес потврде уводи додатне податке (потврду) и додатно кашњење, које је некада вредно тога а некада је непожељно.

Пренос датотека је типична ситуација где је поуздана услуга са успоставом везе неопходна. Власник датотеке жели да буде сигуран да су сви битови тачно примљени и у истом редоследу у коме су и послати. Мали је број корисника који би пристао да дође до мешања или губитка само пар битова, без обзира што би процес преноса био бржи. У неким применам кашњења која се уносе при слању потврде су неприхватљива. Једна таква апликација је пакетизовани говор. За корисника телефона боље је да има шум са линије него уношење кашњења које се добија користећи потврде. Исто је и са преносом пакетизоване живе слике. Постоје апликације којима није потребна успостава веза.

На пример електронска пошта. Потребно је послати поруку и велика је вероватноћа да ће она бити примљена, али се не гарантује. Пакетска порука којом се шаље целокупна информација са адресама пошиљаоца и примаоца назива се датаграм.



Слика 4*. Карактеристике услуга са успоставом и без успоставе везе.*

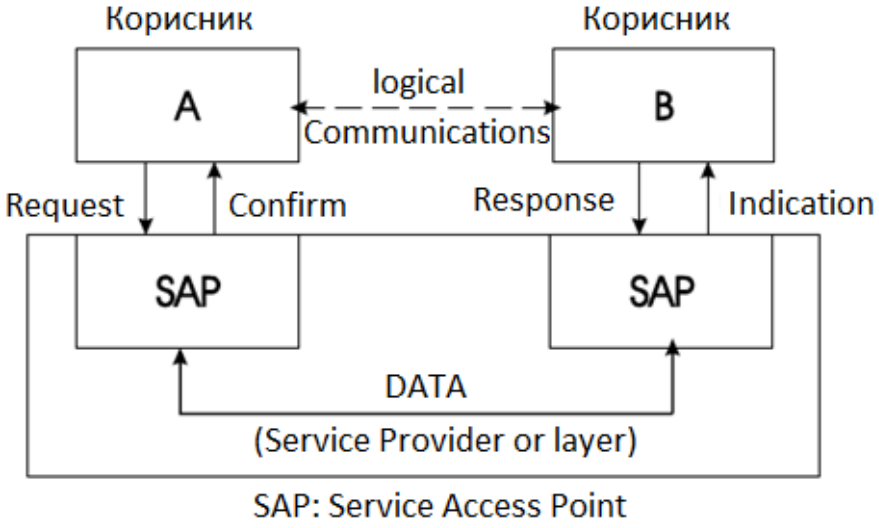
3.2. Основне операције услуга

Услуге се формално специфицира преко скупа операција које су доступне кориснику или другој некој целини и преко којих он приступа услузи. Ове примитиве налажу целини која пружа услугу да изврши неку активност, или да извести о некој активности коју су парњак целине обавиле. Један од начина да се класификују примитиве услуге је да их групишемо у четири класе, као што је приказано у табели 1.

Основне операције	Значење
Request - Захтев	Корисник тражи од целине која нуди услугу да одради неки посао
Indication - Индикација	Корисник је информисан о догађају
Response - Одговор	Корисник жели да одговори на догађај
Confirm - Потврда	Одговор на претходни захтев је стигао назад

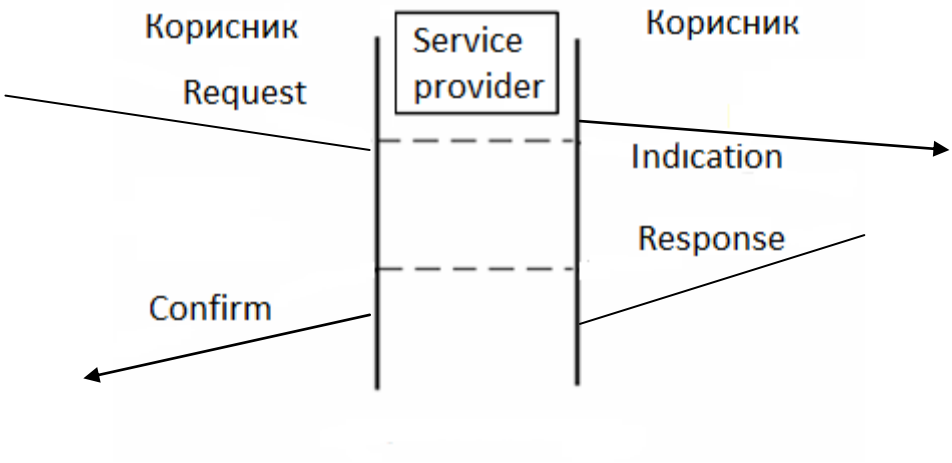
Табела 1. *Четири класе операција услуга.*

Да бисмо илустровали употребу операција, размотримо како се веза успоставља и раскида. Целина која је иницијатор извршава Request, која резултује слањем јединице података протокола. Пријемник добија операцију Indication (обавештење) да нека целина негде жели да успостави са њим везу. Користи Response операцију која указује да ли ће да прихвати или одбаци позив. Целина која је иницирала Request операцију проналази шта се дешава посредством Confirm операције (слике. 5 и 6).



Слика 5. *Успостављање логичке везе.*

Примитиве могу да имају параметре и многе од њих и имају. Параметри за Request операцију могу да специфицирају рачунар са којом треба успоставити везу, тип услуге који се жели и максималну величину поруке која се у вези користи.



Слика 6. *Размена операција.*

Параметар за Indication може да садржи идентитет позивајуће целине, тип услуге који се жели и препоручену максималну величину поруке. Уколико се позвана целина не слаже са препорученом максималном величином поруке, може да направи другачији предлог у Response примитиви, која ће бити доступна пошиљаоцу преко примитиве Confirm. Детаљи договора су део протокола.

3.3. Релација између услуге и протокола

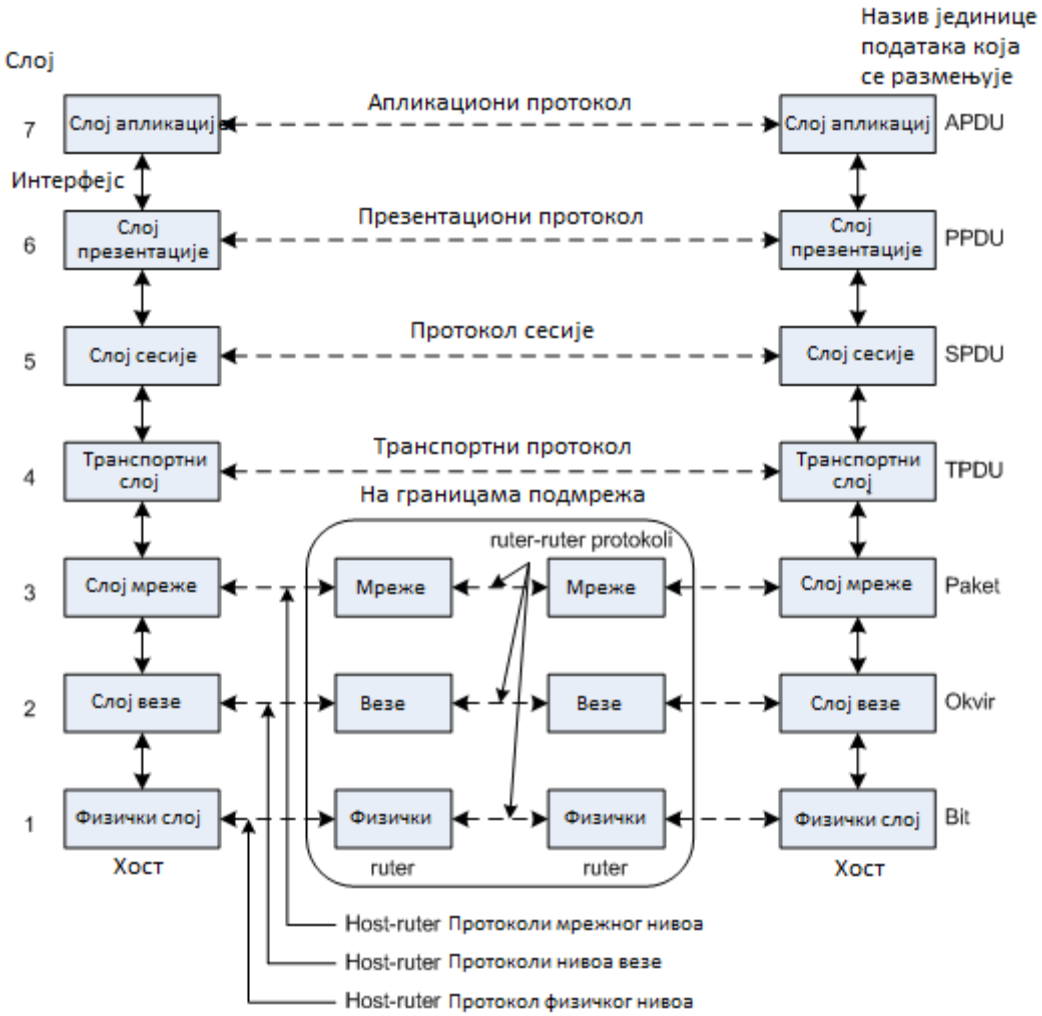
Услуге и протоколи имају различите концепте, који се често мешају. Услуга је скуп примитива (операција) које слој обезбеђује слоју изнад себе. Услуга дефинише које операције је слој спреман да изврши на захтев корисника, али не указује ничим како су ове операције примењене. Услуга је везана за интерфејс између два слоја, са нижим слојем који је давалац услуге и вишим слојем који је корисник услуге.

Протокол, на другој страни, је скуп правила која одређују формат и значење рамова, пакета или порука које се размењују преко парњак целина у оквиру слоја. Целине користе протокол у циљу имплементације дефинисане услуге. Могу слободно да мењају протокол али не могу да мењају услугу која је видљива њиховом кориснику. На овај начин услуга и протоколи потпуно су раздвојени.

Услуга је као апстрактни тип података или објекат у објектно - оријентисаним језицима. Он дефинише операције које треба да се изврше над објектима, али не специфицира како су ове операције имплементиране. Протокол се односи на имплементацију услуге и као таквог га корисник услуге не види.

4. OSI референтни модел

OSI модел приказан је на Сл. 7. *OSI* је стандард уведен 1983. године од стране међународне организације за стандардизацију (*ISO - International Standard Operation*). Његово име, *OSI* (*Open System Interconnect*) референтни модел, указује да се ради о моделу повезивања отворених система, односно система који су отворени за комуникацију са другим системима. OSI није протокол, већ општи модел за разумевање и развој флексибилних, робусних и отворених мрежних архитектура.



Слика 7. *Приказ OSI референтног модела* [6].

OSI модел дефинише седам слојева, чији се називи могу прочитати са Сл. 7. *OSI* не дефинише конкретне протоколе и сервисе које се користе на различитим нивоима, већ описује шта који слој ради. Претпоставка је да се мрежа састоји од већег броја хостова (рачунара) груписаних у више подмрежа повезаних рутерима. Хостови су извори и крајња одредишта информација. У оквиру исте подмреже, хостови директно размењују податке. Размена података између хостова из различитих подмрежа обавља се посредством рутера [4].

Слојеві OSI модела могу се сврстати у три групе:

Слојеви 1, 2 и 3, физички, слој везе и мрежни, су слојеви за подршку рада мреже који се превасходно баве преносом података између хостова (што подразумева, између осталог, спецификацију сигнала, физичких веза и адреса, тајминг и поузданост). Слојеви 1, 2 и 3 не улазе у смисао података који се преносе, већ их третирају као низ бајтова (или битова) које треба поуздано пренети од извора до одредишта.

Слојеви 5, 6 и 7, тј. презентациони, слој сесије и слој апликације, су слојеви за подршку кориснику, који се старају о усклађености презентације података и прописују правила дијалога две удаљене апликације.

Слој 4, транспортни слој, задужен је за успостављање и одржавање конекције и поуздани пренос података између крајњих апликација у мрежи сложене топологије (за разлику од слоја 2 који је задужен за поуздани пренос податаке преко једног линка).

Виши слојеви *OSI* модела (слојеви 4-7) се реализују у софтверу, док су нижи (слојеви 1-3) комбинација хардвера и софтвера, са изузетком физичког слоја који је се увек реализује у хардверу. Слојеви 1, 2 и 3 садржани су у мрежном софтверу и хостова и рутера, док су слојеви 4-7 присутни само у хостовима. У наставку ће бити описана функција сваког слоја.

5. Слојеви OSI референтног модела

Слојеви 1, 2 и 3, физички, слој везе и мрежни, су слојеви за подршку рада мреже који се превасходно баве преносом података између хостова (што подразумева, између осталог, спецификацију сигнала, физичких веза и адреса, тајминг и поузданост). Слојеви 1, 2 и 3 не улазе у смисао података који се преносе, већ их третирају као низ бајтова (или битова) које треба поуздано пренети од извора до одредишта.

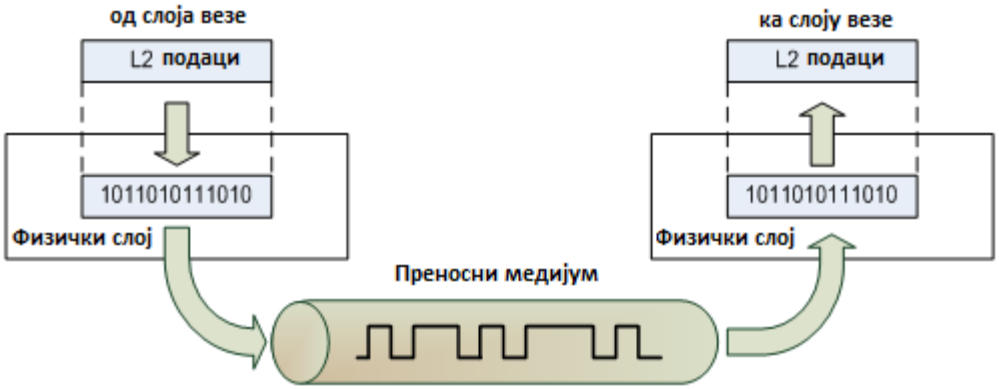
Слојеви 5, 6 и 7, тј. презентациони, слој сесије и слој апликације, су слојеви за подршку кориснику, који се старају о усклађености презентације података и прописују правила дијалога две удаљене апликације.

Слој 4, транспортни слој, задужен је за успостављање и одржавање конекције и поуздани пренос података између крајњих апликација у мрежи сложене топологије (за разлику од слоја 2 који је задужен за поуздани пренос податаке преко једног линка).

Виши слојеви *OSI* модела (слојеви 4-7) се реализују у софтверу, док су нижи (слојеви 1-3) комбинација хардвера и софтвера, са изузетком физичког слоја који је се увек реализује у хардверу. Слојеви 1, 2 и 3 садржани су у мрежном софтверу и хостова и рутера, док су слојеви 4-7 присутни само у хостовима. У наставку ће бити описана функција сваког слоја.

5.1. OSI - Физички слој

Физички слој је одговоран за пренос битова[1] преко физичког преносног медијума (жичаног, оптичког или бежичног линка). Физички слој дефинише механичке и електричне карактеристике програмског медијума и интерфејса[2] између мрежног уређаја и преносног медијума. Такође, дефинише функције и процедуре које уређај и интерфејс треба да спроводе како би се остварио пренос. На слици 5 је приказана позиција физичког слоја у односу на физички преносни медијум и солј везе.



Слика 8. *Позиција физичког слоја у односу на физички преносни медијум.*

Физичке каракетристике интерфејса и медијума. Дефинише тип преносног медијума (Жичани, оптички или бежични) и електричне и механичке карактеристике интерфејса за спегу уредјаја на медијум, све до нивоа типова утичница и распореда пинова на прикључним конекторима.

Репрезентација битова. Подаци на физиком нивоу се састоје од низа битова (непрекидна секвенца 0 и 1-ца). Да би се пренели преко физичког медијума, битови морају на неки начин бити утиснути у сигнал (електрични или оптички). Другим речима, физицки ниво дефинише тип кодирања и модулације (како се битови конвертују у сигнал).

[1] Бит (bit) је најмања јединица информације у рачунарству. Један бит представља количину информације потребну за разликовање два међусобна искључива стања, често представљана као један (1) и нула (0).
[2] Interface обухвата графичке, текстуалне и звучне информације које програм представља кориснику и активности које корисник обавља да би приступио програму (задавање команди тастатуром, померање миша).

Брзина преноса. Изражава број бита који се једној секунди преносе преко физичког медијума (битска брзина), у јединицама као сто су: Kbps (*kilobits-per-scond*) - 210 (= 1024) бита у секунди, или Mbps (Megabits-per-second) 220 (= 1048576 ≈ 1 милиона) бита у секунди. Битски интервал је трајање једног бита и представља реципрочну вредност битске брзине (Сл. 6).

Слика 9. *Битски интервал (трајање једног бита).*

Битска синхронизација. Рад на истој битској брзини није довољан да би пријемник исправно примио битску секвенцу коју шаље предајник. Предајник и пријемник морају бити синхронизовани до нивоа бита. Да би из сигнала издвојио појединачне битове, пријемник мора да има информацију када сваки бит почиње и када се завршава. Физицки ниво дефинише начин на који се остварује синхронизација између предајника и пријемника.

Конфигурација линије. Дефинише да ли се користи *point-to-point*[3] или *multipoint* линијска конфигурација.

Физичка топологија. Дефинише топологију мреже.

Режим преноса. Дефинише смер преноса података између уредјаја (симплекс, полудуплекс, или дуплекс).

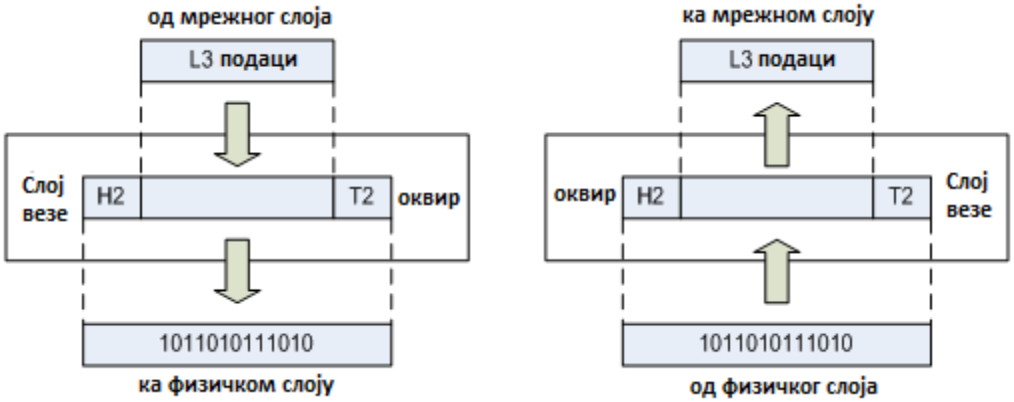
5.2. OSI - Слој везе

Слој везе трансформише физички слој у поуздани линк за испоруку података од чвора до чвора. Користећи сервисе слоја везе, слој мреже види физички слој као идеални преносни медијум у коме се не могу десити грешке које се не могу открити.

Због утицаја разнороврсних поремећаја из окрижења (сметње, шумови, електромагнетска интерференција) у току преноса податак кроз медијум може доћи до нарушавања битксе секвенце.

[3] PPP (Point-to-Point Protocol) користи се за директно повезивање 2 чворова рачунарске мреже. Омогућава повезивање рачунара серијским, телефонским или оптичким каблом.

Тако, може се десити да пријемника погрешно прими неке битове секвенце (1 уместо 0, или обрнуто), или да прими више или мање битова од оног броја који је послат. На слоју везе је да детектује и, ако је то могуће, исправи грешке. Такође, слој везе решава проблем координације брзог предајника и спорог пријемника, као и проблем контроле дељивог физичког медијума (код *multipoint* линкова). Позиција слоја везе у односу на мрежни и физички слој приказана је на Сл 10.



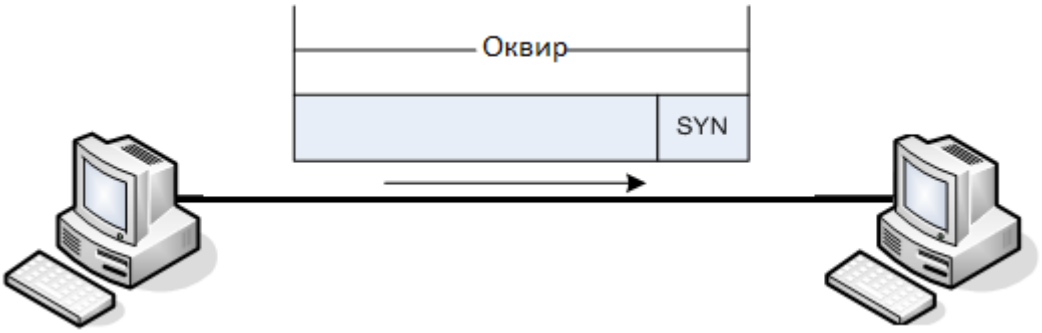
Слика 10. *Позиција слоја везе у односу на мрежни и физички слој.*

Функције OSI слоја веза :

- Уоквиравање
- Физичко адресирање
- Контрола протока

OSI - Слој везе Уоквиравање

Слој везе врши поделу низа битова из слоја мреже на мање јединице које се зову оквири (или фрејмови). Код неких протокола сви оквири морају имати исту, тачно одређену дужину. Код других, дужина оквира може бити променљива, али не већа од максимално дозвољене (типично, неколико стотина до неколико хиљада бајтова). Да би се остварила почетна синхронизација између предајника и пријемника, почетак и крај оквира морају бити једнозначно одређени. По пријему секвенеца за почетак (најчешће облика 1010…1010) пријемник зна да је почео пренос новог оквира.



Слика 11 . *Уоквиравање.*

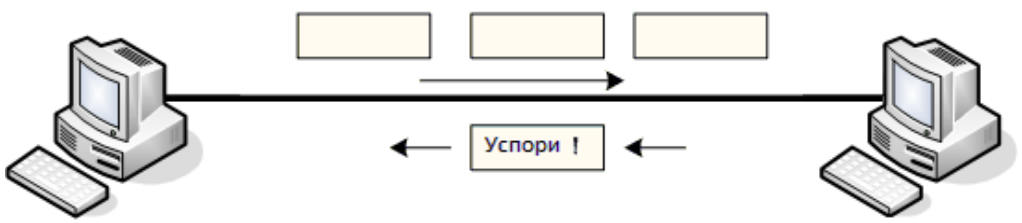
OSI - Слој везе Физичко адресирање

Изузев потпуно повезане мреже, код које је сваки линк искључиво намењен комуникацији између два чвора, код свих осталих мрежних топологија, линкови су дељив ресурс који користе више од два чвора. На тај начин, сигнал који шаље један чвор, иако је намењен само једном одредишном чвору, дистрибуира се до свих чворова у истој подмрежи. Да би се омогућила идентификација одредишта оквира, сваком чвору у мрежи додељује се јединствена физича адреса. Слој везе у заглавље сваког оквира који шаље умеће физичку адресу одредишта и физичку адресу извора поруке. Оквир примају сви чворови у подмрежи, али је прихвата само онај који адресу одредишта препозна као своју адресу. На основу адресе извора, одредишни чвор зна ко је послао оквир. Ако су подаци намењени чвору који се налази у некој другој подмрежи, тј. чвору коме подаци не могу директно да се испоруче, оквири се шаљу на физичку адресу рутера (уређаја који се користи за међумрежно повезивање).

Слика 12 . *Физичко адресирање.*

OSI - Слој везе Контрола протока

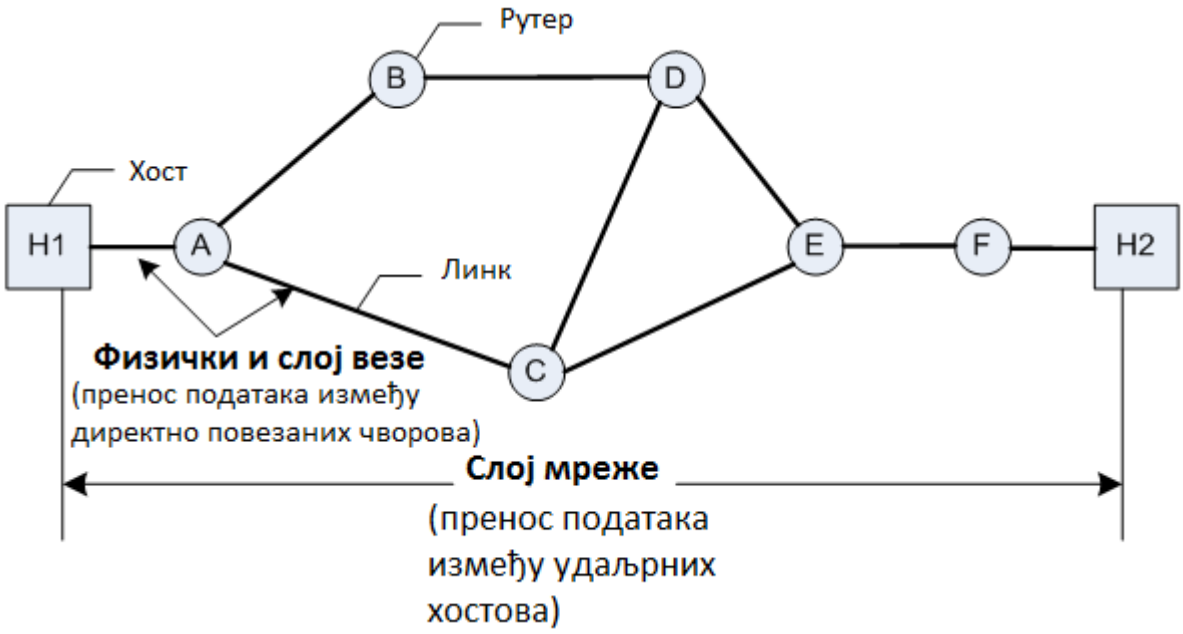
Слој везе поседује механизме који спречавају да пријемник буде ″претрпан″ подацима у случајевима када је брзина којом може да апсорбује податке мања од брзине којом предајник шаље податке. Контрола протока заснована је на повратним оквирима које пријемника шаље предајнику, а којима му налаже да привемено обустави, односно настави слање нових оквира. На пример, пријемник може да пошаље предајнику поруку следећег значења: ″Можеш да ми пошаљес н оквира, али после тога престани са слањем и чекај док ти не кажем када да наставиш.″



Слика 13. *Контрола протока.*

5.3. OSI – Слој мреже

Слој мреже одговоран је за испоруку пакета од извора до одредишта који се могу налазити и у различитим мрежама (нису повезани на исти линк). Ако су два система повезана на исти линк, обично не постоји потреба за мрежним слојем. Међутим, ако су системи повезана на различите мреже (линкове), са уређајем за међумрежно повезивање између њих, мрежни ниво је неопходан, а његов задатак је регулација протока пакета између два система. На Сл. 14 је приказан однос између слоја мреже, транспортног слоја и слоја везе.



Слика 14. *Однос слоја мреже, транспортног слоја и слоја везе.*

Када пекети прелазе границе подмрежа, могу настати бројни проблеми. Физичко адресирање које се користи у другој мрежи се може разликовати од онога које важи у првој. Пакет који стиже из једне подмреже може бити превише велики да би се у другој мрежи пренео једним оквиром. Могу се разликовати протоколи нижег нивоа. На слоју мреже је да реши све ове проблеме.

Функције OSI - Слоја мреже:

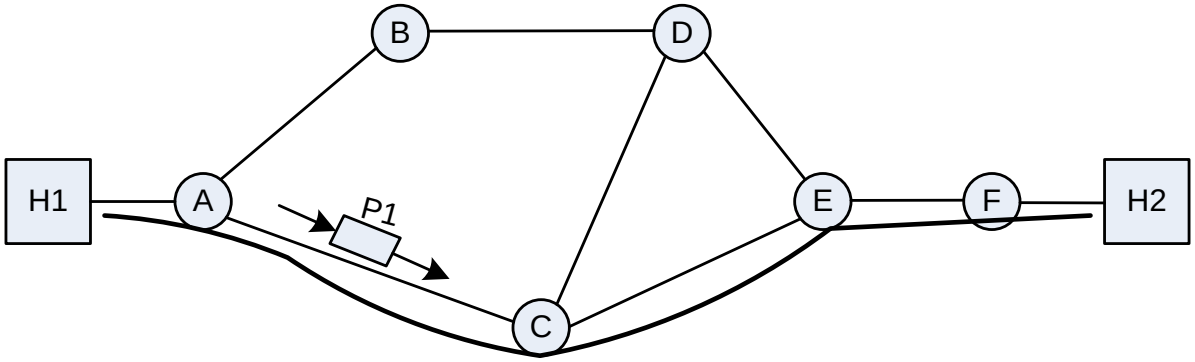
- Логичко адресирање
- Рутирање
- Фрагментација пакета
- Контрола загушења

OSI – Слој мреже Логичко адресирање

Физичко адресирање, које се реализује на нивоу слоја везе, решава проблем адресирања локано, на нивоу заједничког линка. Сложена мрежа, формирана повезивањем више, могуће различитих подмрежа, које користе различите шеме физичког адресирања, захтева увођење логичких (или мрежних) адреса, које ће бити јединствена на нивоу целокупне мреже

OSI – Слој мреже Рутирање

На Сл. 15 је илустрован контекст у коме слој мреже ради. Круговима су означени рутери, а квадратима хостови. хост X1, шаље пакет удаљеном хосту X2. Рутери повезани преносним линијама формирају ″костур″ велике мреже.



Слика 15 . *Принцип по коме слој мреже ради.*

У систему постоји још велики број хостова, распоређених у локалне мреже које се могу формирати око сваког рутера, али ради једноставности нису приказани.

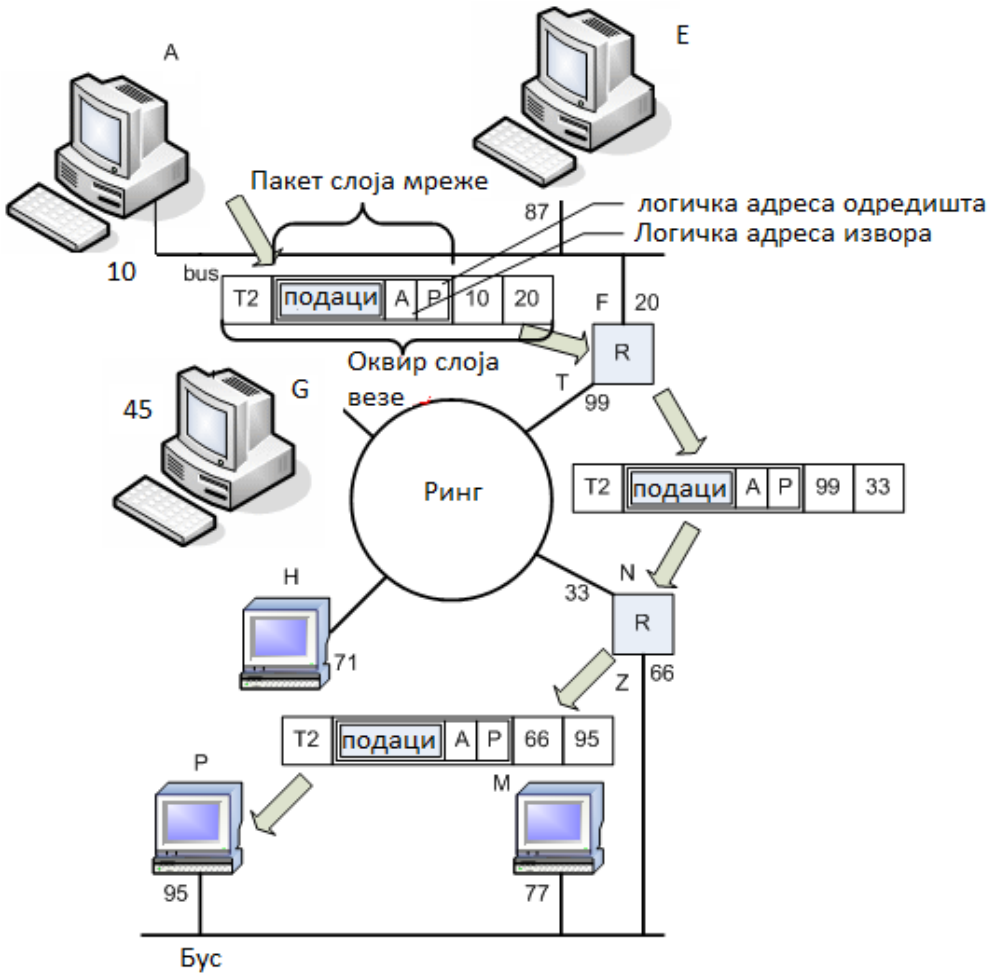
У сваком случају, сваки хост у систему може директно да комуницира са барем једним рутером, а рутери су тако међусобно повезани да се између сваког пара удаљених хостова може успоставити веза.

Другим речима, између свака два удаљена хоста у систему постоји једна или више путања која се може формирати надовезивањем рутера и линкова између њих. Путању којом пакет П1 послат од стране хоста X1 стиже до хоста X2 чине рутери A, C, E и F. Приметимо да је пакет P1 могао бити испоручен хосту X2 и неком другом путањом: (A, B, D, E, F) или (A, C, C, E, F) или чак (A, B, D, D, E, F).

Задатак протокола соја мреже, који се извршава како у хостовима, тако и у рутерима, је да на основу логичке адресе одредишта из заглавља примљеног пакета одлучи којим линком пакет треба да продужи даље ка свом крајњем одредишту.

Мрежу чине три подмреже (две типа бус и једна типа ринг, повезане рутерима који су означени симболом P. Приметимо да две бус мреже нису међусобно директно повезане, што значи да сваки пакет који се размењује између њих мора проћи кроз ринг мрежу у средини. Нека чвор са мрежном адресом A и физичком адресом 10, лоциран у једној бус мрежи, шаље пакет чвору са мрежном адресом P и физичком 95 из друге бус мреже. С обзиром да се два чворова налазе у различитим подмрежама,

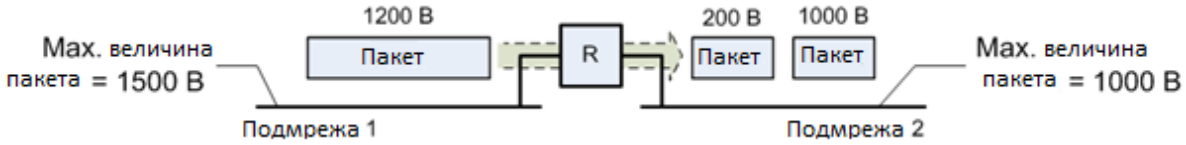
физичке адресе нису довољне; физичке адресе имају само локално значење. Неопходне су универзалне адресе које ће важити и изван граница локалних мрежа. Ту особину имају мрежне (логичке) адресе. Ток преноса пакета од чвора А до чвора Р, одвија се на следећи начин: Чвор А закључује да се чвор Р, коме жели да пошаље пакет, не налази истој подмрежи и зато пакет шаље локалном рутеру (F). Преко директног линка пакет, уоквирен у оквир слоја везе, се шаље на физичку адресу рутера (20). По пријему пакета, рутер F на основу логичке адресе одредишта пакета закључује да пакет не може директно да испоручи чвору Р, и зато га прослеђује рутеру Н. Конкретно, пакет се шаље на физичку адресу рутера Н (66). Рутер Н провером логичке адресе из примљеног пакета закључује да је пакет намењен чвору који је њему директно доступан, Р, и због тога, пакет, шаље директно одредишном чвору (упакован у оквир послат на физичку адресу чвора Р, 95). Уочимо да пакет који се на нивоа слоја мреже преноси између чворова А и Р, садржи логичке адресе извора, А, и одредишта, Р, које остају неизмењене на целом путу кроз мрежу; мењају се физичке адресе када пакет прелази из једне у другу подмрежу. Уочимо, такође, да за разлику од хостова који имају једну физичку и једну логичку адресу, рутери имају два, или у општем случају више, физичких и логичких адреса које су јединствене за сваку мрежу на коју је рутер прикључен.



Слика 16 . *Комуникације на даљину.*

OSI - Слој мреже Фрагментација пакета

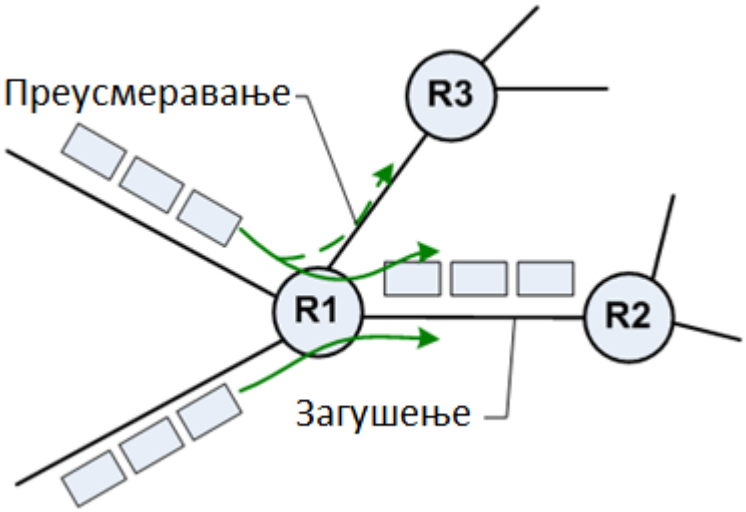
Подмреже повезане у интернет намећу различита ограничења у погледу максималне величине пакета које могу да преносе. Ова ограничења су директна последица ограничене величина оквира слоја везе која зависи од типа физичког медијума који се користи у подмрежи. Настаје проблем када велики пакет треба да пређе у подмрежу код које је максимална величина пакета исувише мала да би пакет могао бити пренет ″у једном комаду″. Да би се овај проблем превазишао, рутерима је дозвољено да велике пакете деле на више мањих фрагмената и сваки фрагмент преносе даље као независни пакет. На одредишту, фрагменти се прикупљају и спајају у првобитне пакете. Наравно, нека форма нумерисања фрагмената је неопходна. Фрагментација и реконструкција пакета су у надлежности слоја мреже.



Слика 17. *Фрагментација пакета.*

OSI - Слој мреже Контрола загушења

Ако се у мрежи, у исто време, налази велики број пакета, који се кроз рутере и линкове, преносе ка својим одредиштима, перформансе мреже могу значајно да деградирају. Оваква ситуација се зове загушење. Загушењем су обично погођени поједини делови мреже. Пренос пакета кроз загушене рутере и/или линкове се успорава, а у условима веома интензивног саобраћаја, поједини пакети могу бити и изгубљени. Контрола загушења је одговорност мрежног слоја, који треба да преусмери саобраћај из загушених делова мреже ка рутерима који имају мање посла.



Слика 18. *Контрола загушења.*

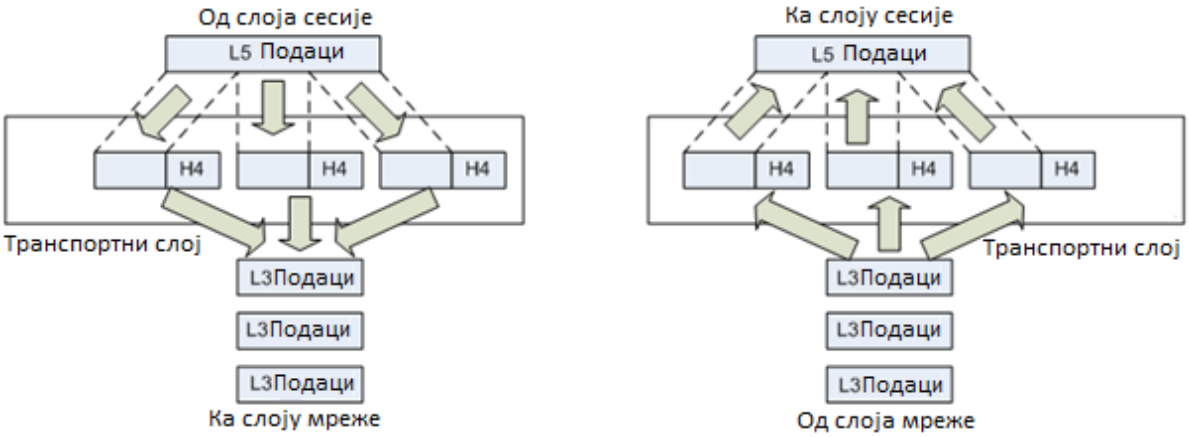
5.4. OSI – Транспортни слој

Транспортни слој је одговоран за испоруку целокупне поруке од извора до одредишта (тј. од-краја-до-краја). Мрежни слој иако обезбеђује пренос појединачних пакета од извора до одредишта, не ″види″ било какву везу између њих, већ сваки пакет третира као независну јединицу; као да је сваки пакет посебна порука, без обзира да ли је то и заиста случај или не.

Такође, слој мреже, иако чине навише шта може, не гарантује да ће сваки пакет бити испоручен. Што више, ако пакет буде изгубљен, нпр. због загушења рутера, мрежни слој никога неће обавестити о томе. Са друге стране, транспортни слој обезбеђује да целокупна порука, у изворном облику, буде пренесена до одредишта, намећући контролу грешака и контролу протока на нивоу извора и одредишта. На пример, фајл трансфер апликација има задатак да фајл произвољне величине пренесе од фајл сервера на хост који је тражио фајл. У циљу преноса кроз мрежу, фајл ће бити подељен на пакете, а сваки пакет ће се преносити независно.

Неки пакете могу бити примљени са грешком, а неки изгубљени у преносу. Задатак транспортног слоја је да уведе строгу дисциплину у испоруци пакета као би фајл у првобитном облику био пренет до свог одредишта. На Сл. 19 је приказан однос између транспортног слоја и слојева мреже и сесије. Основна функција транспортног слоја је да прихвати податке од вишег слоја, подели их на мање јединице, ако је то потребно, проследи их слоју мреже и осигура да ће сви они коректно стићи на други крај. Додатно, све то мора бити обављено ефикасно и на начин који ће изоловати више слојеве од евентуалних промена на нижим слојевима (условљених рецимо променом хардвера мреже).

Суштинска разлика између транспортног и слојева нижег нивоа је у томе што се ниже нивои баве комуникацијом између машине и њених непосредних суседа, док транспортни слој подразумева комуникацију између крајњих машина, које могу бити раздвојене већим бројем рутера.

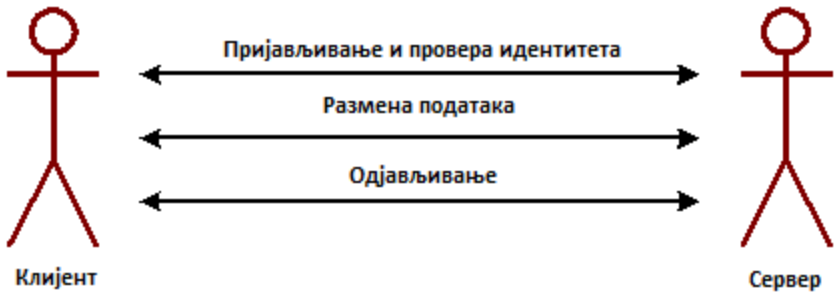


Слика 19 . *Однос између транспортног слоја и слојева мреже и сесије.*

5.5. OSI – Слој сесије

Слој сесије омогућава корисницима на различитим машинама да успоставе сесију између њих. Сесија пружа различите сервисе, као што су: управљање дијалогом (ко и када може да шаље податке), контрола приступа заједничким ресурсима (да би се спречило да две стране у исто време покушају извођење неке критичне операције) и синхронизација (надгледање дуготрајних преноса великих фајлова за случај абнормалног прекида како би се по поновном успостављању комуникације пренос наставио почев од тачке прекида).

Сервис за управљање дијалогом:



Слика 20. *Управљање дијалогом.*

Клијент има утисак да комуницира са једим сервером преко једне сесије, иако:

Сесија може да укључи више конекција и интеракцију са различитим серверима. Конекција може непредвиђено да се прекине, а задатак сесије је да је поново успостави.

У многим случајевима, интеракција две удаљене апликације не укључује само просту размену података. Обично, једна страна у комуникацији има улогу клијента (онај ко тражи услугу), а друга сервера (онај ко пружа услугу). Да би сервер опслужио клијента, клијент најпре мора да се представи и на неки начин докаже свој идентитет (нпр. путем корисничког имена и лозинке), како би сервер био сигуран да клијент има право коришћења тражене услуге или ресурса из тог система.

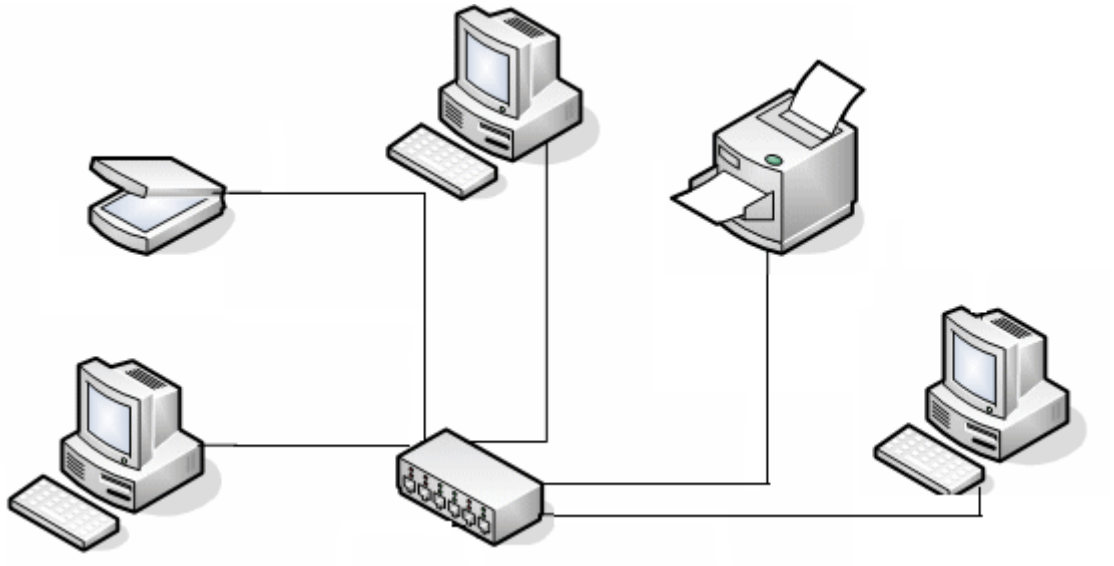
По завршетку интеракције, клијент се одјављује. Све ове активности чине дијалог између два система који се остварује под контролом слоја сесија. У основи, дијалог може бити типа полудуплекс (наизменично у једну и другу страну) или пуни дуплекс (истовремено у обе стране). Слично конекцији са транспортног нивоа, сесија се отвара, траје и затвара.

Међутим, једна сесија може да укључи већи број конекција. На пример, свака фаза сесија може захтевати посебну конекцију на транспортном нивоу. Конекција може непредвиђено да се прекине, а задатак слоја сесије је да конекцију поново отвори. Шта више, сервер не мора бити један рачунар, већ један рачунар може бити задужен за пријављивање и ауторизацију, док други може садржати базу података.

Слој сесије све ове детаље сакрива од клијента, који има утисак да комуницира са јединственим сервером преко јединствене сесије.

Сервис за контролу приступа заједничким ресурсима:

- Обезбеђује узајамно искључиво коришћење заједничких ресурса
- Заједнички ресурси: база података, штампач ...



Слика 21. *Узајамно коришћење дељивих мрежних ресурса.*

Слој сесије омогућава безбедно и узајамно искључиво коришћење дељивих мрежних ресурса. На пример, ако је ресурс мрежни штампач, јасно је да у једном времену сервис штампања може да опслужује само једног клијента. Другим речима, увек може да буде отворена највише једна сесија штампања. Такође, дељив ресурс може бити база података којој приступа велики број корисника.

Сервис за синхронизацију:

Слој сесије омогућава процесима да у ток података умећу синхронизационе тачке. На пример, ако неки систем шаље фајл од 2000 страница, има пуног смисла постављати синхронизационе тачке на сваких 100 пренетих страница како би се потврдило да је свака од јединица обима 100 страница успешно пренета и потврђена од стране одредишта. Ако којим случајем у току преноса 535-те странице дође до непредвиђеног прекида комуникације, нпр. због нестанка напајања, након поновног успостављања везе ретрансмисија ће почети од 501 странице, јер странице од 1 до 500. нема потребе поново преносити. Такође, синхронизационе тачке приликом спровођења неког сложеног дијалога и се убацују након окончаних појединачних фаза дијалога. Ако усред дијалога дође до прекида комуникације, дијалог ће бити настављена почев од последње комплетиране фазе.

5.6. OSI – Презентациони слој

Презентациони слој је задужен за синтаксу и семантику информација које се размењују између два система. У слоју презентације се обављају трансформације података, које су неопходне како би се ускладили формати података, омогућило рационално коришћење комуникационог капацитета мреже и обезбедила сигурност података.

Превођење:

Процеси (програми који се извршавају) на два удаљене система обично размењују информације у облику низова карактера, бројева и тд. Пре преноса, информација мора бити конвертована у низ битова. Различити рачунари користе различите системе кодирања, а одговорност презентационог слоја је да обезбеди превођење из једног у други.

На страни предаје, презентациони слој преводи информацију из формати који користи пошиљалац у неки стандардни формат, разумљив свима. На пријемној страни, презентациони слој преводи информацију из стандардног у формат који користи одредишни корисник. На пример, за бинарно кодирање текстуалних информација (слова, цифре и специјални знаци) у употреби су два стандарда: *ASCII* и *EBCDIC*. Слово ´A´, у *ASCII* стандарду кодира се са 7 бита као ¨´´´, а у *EBCDIC* са 8 бита као ´´´. Очигледно текстуални фајл креиран на машини која користи *ASCII* стандард, биће нечитљив на машини где је у употреби *EBCDIC* стандард, осим ако се пре испоруке текста крајњем кориснику не обави превођење из ASCII у *EBCDIC*.

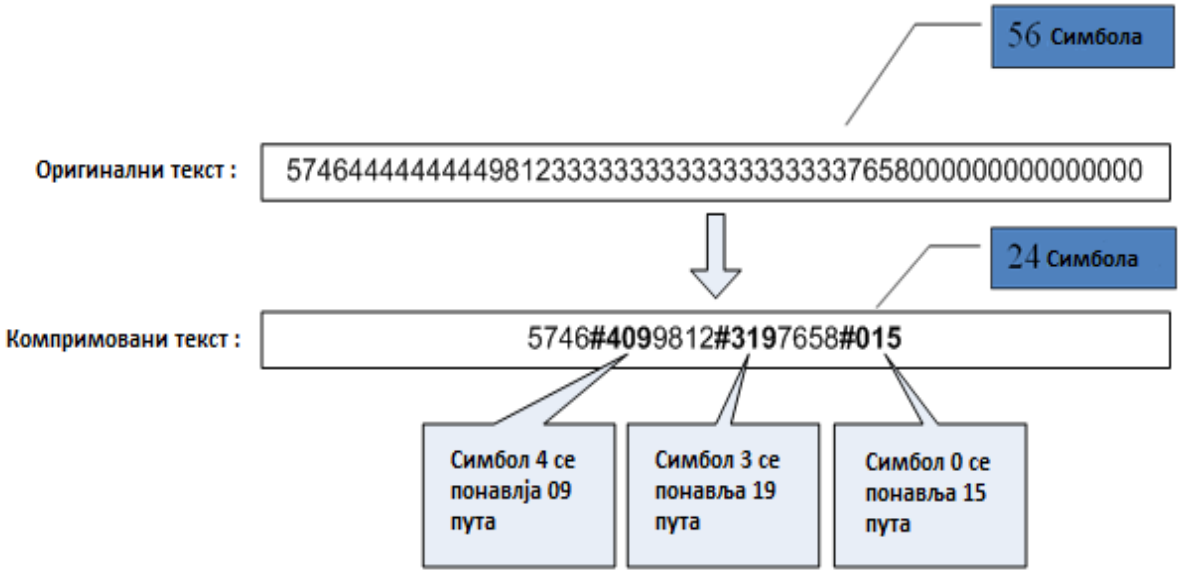
Компресија / Декомпресија:

Између рачунара често се преносе обимни фајлови које садрже текст, слике или неки друге типове података. Величина ових фајлова може бити више десетина или чак стотина мегабајта. Пренос тако великих фајлова траје дуго и заузима значајан део капацитета мреже. Да би се скратило време преноса, на предајној страни се обавља компресија фајла, чиме се њена величина смањује. Компримовани фајл се преноси преко мреже до одредишног рачунара, где се обавља декомпресија и фајл, у обновљеном облику, испоручује одредишној апликацији.

Подаци се компримују уклањањем низова битова који се понављају и њиховом заменом неком врстом прегледа који заузима мање простора; подаци се декомпримују реконструкцијом понављаних секвенци битова. За датотеке које садрже текст, програм или нумеричке податке морају се користити методи компресије без губитака. Процес компримовања без губитака је такав да садржај датотеке после компресије и декомпресије остаје неизмењен, чак и на нивоу бита.

Такви методи обично смањују датотеку на до 40% њене оригиналне величине. Процес компримовања током кога се губе неки подаци зове се компресија са губицима.

Видео и аудио датотеке често садрже више информација него што посматрач или слушалац може да опази; метод компресије са губицима не чува вишак информација и може да смањи ове датотеке на до 5% оригиналне величине.



Слика 22. *Компресија текста.*

Компресија текста, низови идентичних симбола замењују се секвенцином - (#, симбол, број појављивања).

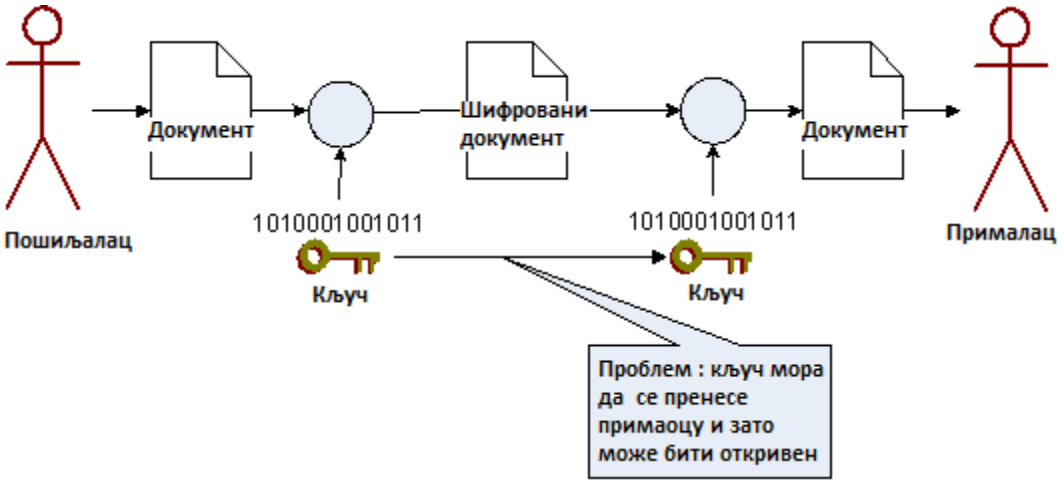
Шифровање / Дешифровање:

Шифровање се односи на заштиту података од неовлашћеног коришћења. Многе рачунарске мреже (као што је Интернет) су јавне мреже, што значи да су доступне свим заинтересованим корисницима. То такође значи да комуникациона инфраструктура која повезује два удаљена корисника није под контролом ни једног од њих, па информације које се размењују могу бити доступне и некој трећој страни и евентуално злоупотребљене. Да би се то спречило, поверљиве информације које се шаљу, на предајној страни се шифрују и преносе у шифрованом облику. На пријемној страни, обавља се дешифровање и информација у првобитном облику испоручује примаоцу. За било које треће лице које дође у посед шифроване информације, она нема значај јер нико осим пошиљаоца и примаоца није у стању да протумачи њено значење.

У случајевима када постоји много пошиљаоца и један прималац, проблем са кључем шифре се превазилази коришћењем асиметричне шеме шифровања која се зове шифровање јавним кључем, а која користи пар кључева: јавни кључ шифрује податке, а одговарајући тајни или приватни кључ их дешифрује. Прималац тајних порука креира јавни кључ и дистрибуира га свим заинтересованим корисницима. Сви корисници користе исти алгоритам шифровања и исти јавни кључ. Међутим, алгоритам дешифровања и кључ дешифровања су тајни и не напуштају примаоца. Метод је асиметричан зато што користи различите алгоритме и кључеве за шифровање и дешифровање. Алгоритам дешифровања је тако осмишљен да није инверзан алгоритму

шифровања. То значи да ако неко дође у посед шифрованих података, чак иако зна алгоритам шифровања и јавни кључ којим је шифровање обављено, неће бити у стању (барем не лако) да реконструише алгоритам дешифровања и одговарајући тајни кључ.

Шифровање помоћу кључа: Шифровање се обично заснива на кључу шифре који је битан за дешифровање. Кључ шифре је низ података који се користи за шифровање других података и који, према томе, мора да се користи и за дешифровање података. Проблем са кључем шифре је што кључ шифре мора да се пренесе примаоцу, што омогућава његово откривање.



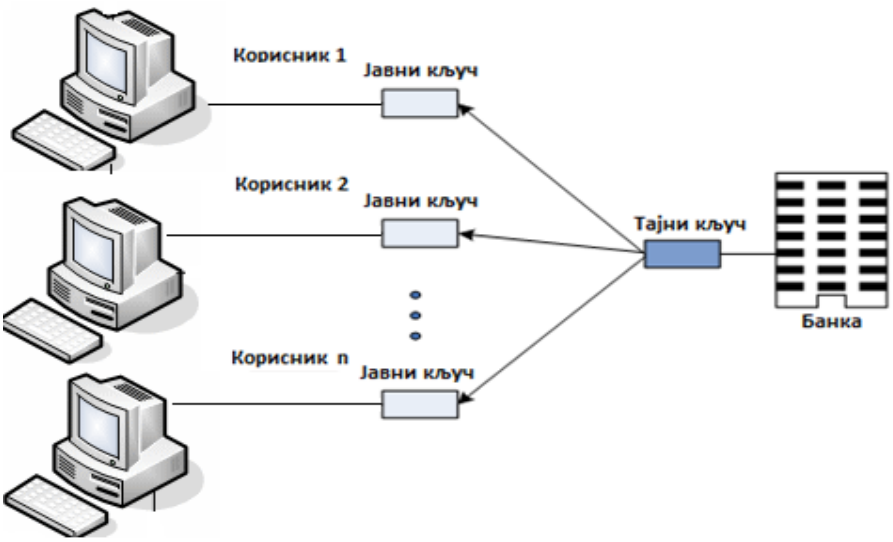
Слика 23. *Преношење кључа примаоцу ради дешифровања документа.*

У случајевима када постоји много пошиљаоца и један прималац, проблем са кључем шифре се превазилази коришћењем асиметричне шеме шифровања која се зове шифровање јавним кључем, а која користи пар кључева: јавни кључ шифрује податке, а одговарајући тајни или приватни кључ их дешифрује.

Прималац тајних порука креира јавни кључ и дистрибуира га свим заинтересованим корисницима. Сви корисници користе исти алгоритам шифровања и исти јавни кључ. Међутим, алгоритам дешифровања и кључ дешифровања су тајни и не напуштају примаоца.

Метод је асиметричан зато што користи различите алгоритме и кључеве за шифровање и дешифровање. Алгоритам дешифровања је тако осмишљен да није инверзан алгоритму шифровања.

То значи да ако неко дође у посед шифрованих података, чак иако зна алгоритам шифровања и јавни кључ којим је шифровање обављено, неће бити у стању (барем не лако) да реконструише алгоритам дешифровања и одговарајући тајни кључ.



Слика 24. *Шифровање јавним кључем на примеру банке.*

На Сл. 24 је илустрована идеја шифровање јавним кључем на примеру банке која корисницима омогућава приступ својим сервисима. Банка је свим својим корисницима проследила исти, јавни алгоритам и кључ шифровања. Тајни кључ и алгоритам, који омогућавају дешифровање поседује само банка.

Провера аутентичности (authentication):

Се односи на утврђивање пуноважности информације. Постоји више метода за проверу аутентичности, али размотрићемо само један, познат под називом дигитални потпис. Концепт дигиталног потписа сличан је процедури која се спроводи приликом нпр. подизања веће количине новца у банци. Корисник потврђује пријем новца потписивањем одговарајућег документ, а банка смешта потписани документ у архиву.

Потписани документ је потребан као доказ о обављеној трансакцији, за случај да се корисник касније обрати банци са тврдњом да није узимао новац. Другим речима, стављањем потписа на документ корисник потврђује аутентичност документа.

Слика 25. *Утврђивање пуноважности информације.*

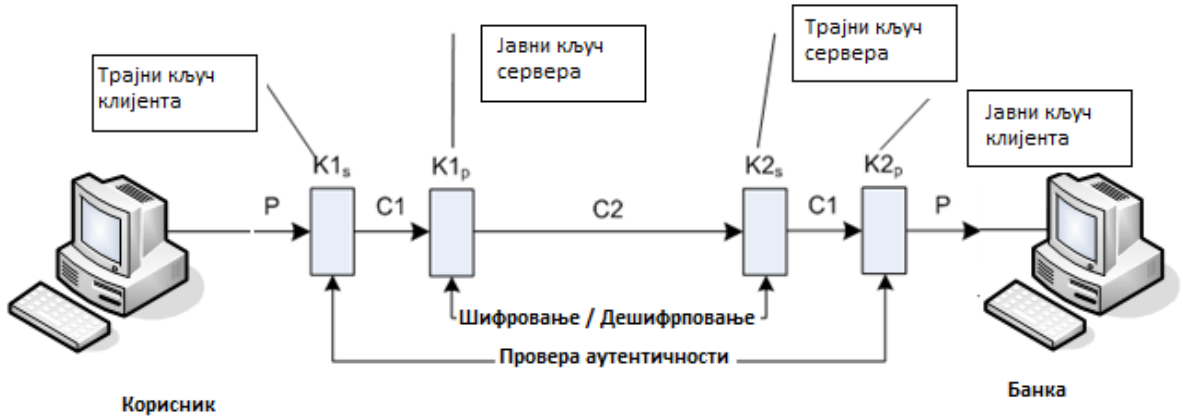
Дигитални потпис:

У мрежним трансакцијама, корисник не може лично да потпише захтев за подизање новца. Уместо тога, корисник може да креира еквивалент електронског или дигиталног потписа начином на који шаље податке. Дигитални потпис уводи још један ниво шифровања/дешифрована у претходно описани процес шифровања јавним кључем. Међутим, овог пута, тајни кључ чува корисник, а јавни банка. Концепт је илустрован на слици.

Корисник шифрује поруку P својим тајним кључем K1p и тако креира прву верзију кôдираног текста поруке (C1). Коришћењем јавног кључа K1п корисник шифрује кôдирани текст C1 и добија другу кôдирану верзију текста поруке (C2) коју шаље банци. Банка корисни тајни кључ K2s да би дешифровала C2 у C1. Верзију поруке C1 банка смешта у своју базу података (архиву), а затим примењује јавни кључ K2p да би од C1 дошла до текста поруке у првобитном облику, P. Текст може да гласи: ″Ја Петар Петровић са свог рачуна бр. тај-и-тај подижем 100 динара″. Ако једног дана Петар Петровић дође у банку са тврдњом да никада није обавио овакву трансакцију, банка може да из своје архиве узме C1, примени K2p и тако покаже Петру да је управо он креирао П. Кључ K2п који дешифрује C1 у P, доказ је да је P шифровано у C1 кључем K1c. Петар не може да тврди да је банка креирала C1, јер банка не поседује K1s; K1s је тајни кључ корисника који једино он зна.

Петар може да се жали суду да је неки неауторизовани корисник дошао у посед кључа K1s. Али, у том случају, суд ће одбацити жалбу са образложењем да је обавеза сваког корисника да свој тајни кључ чува на тајном месту.

Напоменимо да треба правити разлику између провере овлашћења, која се спроводи на нивоу сесије и провере аутентичности са презентационог нивоа. Проверу овлашћења спроводи систем (сервер) да би клијенту оборио коришћење своји ресурса или података (путем захтева за корисничким именом и лозинком). Проверу аутентичности користи клијент да би потврдио аутентичност својих података. Провера овлашћења се спроводи кроз дијалог клијента и сервера на почетку сесије. Провера аутентичности је уграђена у саму информацију коју клијент шаље серверу.



Слика 26. *Провера овлашћења и аутентичности.*

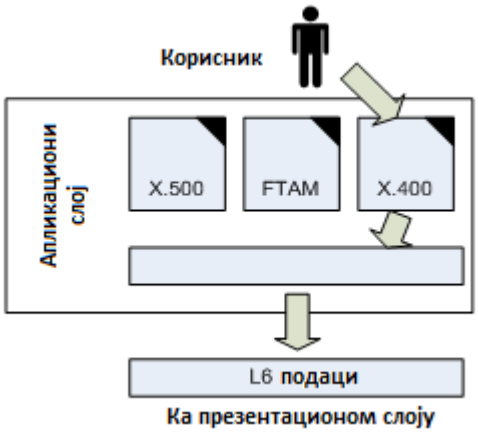
5.7. OSI – Апликациони слој

Слој апликације је вршни слој *OSI* модела који омогућава кориснику коришћење услуга мреже. Сврха шест нижих слојева је обезбеђивање поузданог преноса података. Међутим, пренос података, сам по себи, није крајњи циљ. Тек на апликационом нивоу, могућност размене података са удаљеним корисницима се уобличава у сврсисходне сервисе и апликације. Слој апликације обезбеђује интерфејс и подршку за стандардне сервисе као што су електронска пошта, приступ и пренос удаљених фајлова, Веб и др. Напоменимо да корисник не мора бити човек. Корисник може бити нека друга апликација која се извршава на истом рачунару. У том случају, интерфејс нису тастатура, миш и екран већ скуп функција (сервис) које су на располагању корисничком програму.

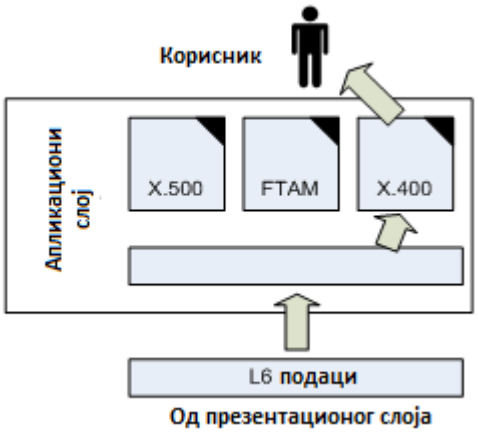
Да би две апликације могле да комуницирају неопходно је да постоје правила која дефинишу скуп дозвољених порука и активности које програм предузима по пријему поруке.

На пример, програм за слање електронске поште омогућава кориснику да напише е-маил, наведе одредишну е-маил адресу и простим кликом на дугме пошаље е-маил. Задатак програма је да садржај писма, адресу пошиљаоца, адресу примаоца заједно другим пратећим информацијама упакује у поруку која ће бити разумљива за програм који користи прималац писама, а да затим успостави везу са Маил сервером и испоручи му поруку. Дакле, оно што за крајњег корисника представља једноставну активност, програм разлаже на читав низ акција које укључују интеракцију и дијалог са неком удаљеном апликацијом.

Да би две апликације могла да се разумеју неопходно је да обе поштују нека заједничка стандардизована правила. Управо правила интеракције између удаљених апликација представљају протоколе који спадају у слој апликације.



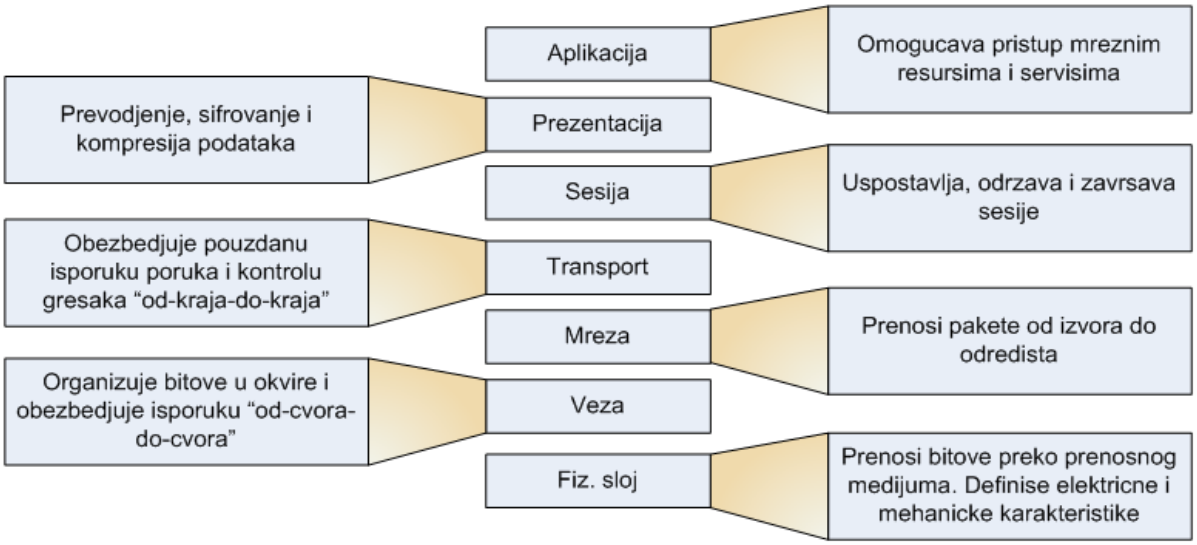
Слика 27. *Однос корисника ка презентационом слоју.*



Слика 28. *Однос корисника од презентационом слоју.*

На Слици. 27 је приказан однос између корисника, слоја апликације и презентационог слоја. Од бројних апликационих сервиса, Слика. 28 приказује само три: X.400 (сервиса за размену података), X.500 (сервис директоријума) i *FTAM* (пренос приступ и менаџмент фајлова). (Напоменимо да се ради о сервисима *OSI* стандарда, који се данас ретко користе.).

OSI – Преглед функција



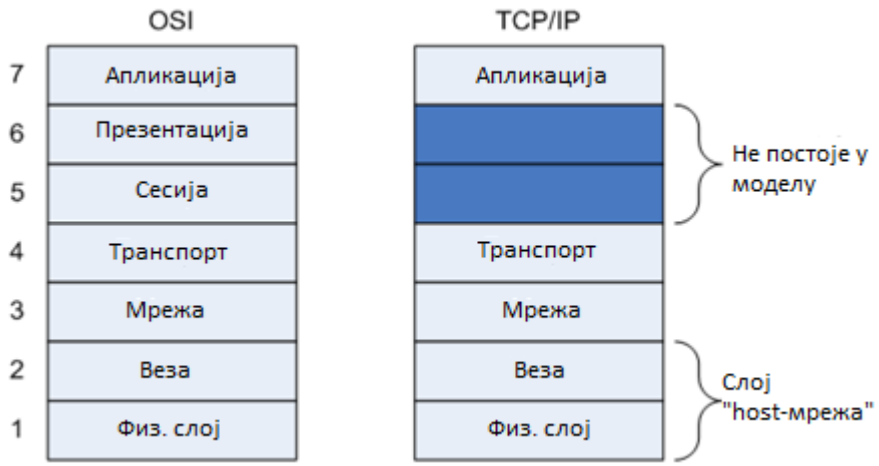
Слика 29 *. OSI – Преглед функција [7].*

6. TCP/IP референтни модел

TCP/IP (*Transmission Control Protocol/Internet Protocol* - Протокол за контролу преноса/Интернет протокол) референтни модел се користи на Интернету. Развијен је пре *OSI* модела, тако да се слојеви ова два модела не поклапају у потпуности. *TCP/IP* модел чини пет слојева:

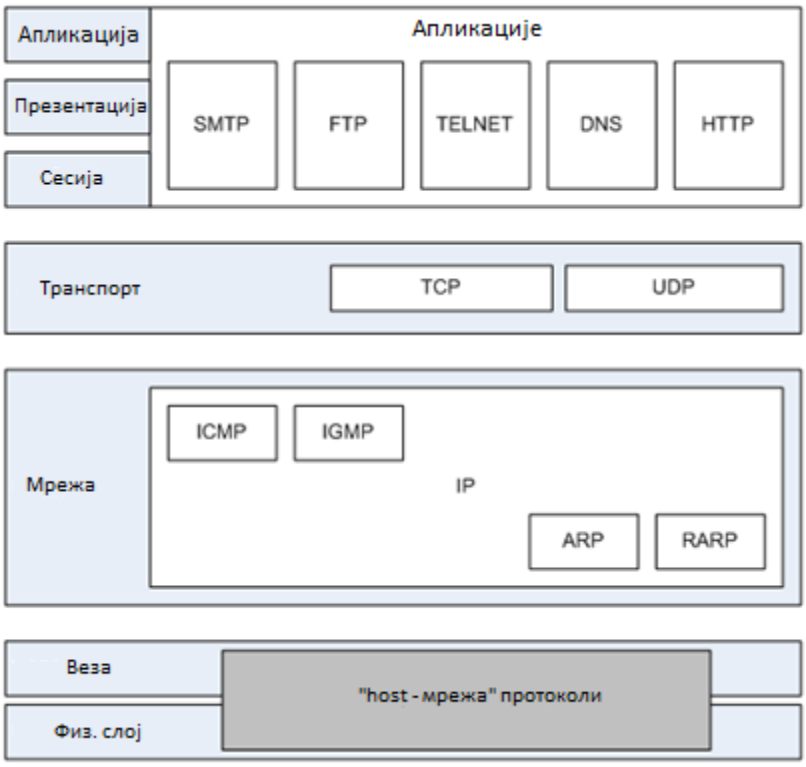
физички, слој везе, мрежни, транспортни и апликациони. *TCP/IP* се само спорадично бави најнижим слојевима (физичким и слојем везе). Заједно, ова два слоја се третирају као ″хост-мрежа ″слој. *TCP/IP* не намеће неке посебне захтеве који се тичу ових слојева (претпоставља се да мрежа поседује протоколе који покривају функције тих слојева), а нагласак ставља на слој мреже, транспортни и апликациони слој.

Мрежни и транспортни слој одговарају слојевима 3 и 4 *OSI* модела. Међутим, код *TCP/IP* на транспортни слој директно се наставља апликациони слој, који обухвата функционалност три вршна слоја *OSI* модела



Слика 30. *TCP/IP референтни модел.*

TCP/IP је хијерархијски протокол сачињен од интерактивних, али не обавезно и међусобно независних модула од којих сваки остварује неку специфичну функцију. За разлику од *OSI* модела који дефинише које функције припадају ком слоју, слојеви *TCP/IP* модела садржи релативно независне протоколе који се могу комбиновати зависно од потреба система. Појам хијерархијски значи да је сваки протокол вишег нивоа, подржан од стране једног или више протокола нижег нивоа. На Сл. 31 је приказана структура *TCP/IP* модела са протоколима разврстаним у слојеве који су преклопљени са одговарајућим слојевима *OSI* модела.



Слика 31. *Структура TCP/IP модела са протоколима разврстаним у слојеве који су преклопљени са одговарајућим слојевима OSI модела.*

6.1. TCP/IP - Мрежни (Интернет) слој

Главни протокол на мрежном нивоу је *IP* (*Internet Protocol*). Поред *IP*, слој мреже садржи још неколико помоћних протокола (*ARP, RARP, ICMP, IGMP* и др.). Интернет слој је одговоран за испоруку пакета од хоста до хоста на Интернету. Главна брига овог слоја је рутирање пакета и избегавање загушења (одговара мрежном слоју OSI модела).

6.2. TCP/IP - Транспортни слој

На транспортном нивоу, *TCP/IP* дефинише два протокола: *TCP* и *UDP* (Усер Датаграм Протоцол). *TCP* је транспортни протокол конекционог типа који омогућава успостављање поузданог тока бајтова између две удаљене апликације. *TCP* обавља сегментацију тока бајтова на поруке које прослеђује интернет слоју. На страни одредишта, *TCP* реконструише ток бајтова и прослеђује га апликацији. Уз то, *TCP* се бави контролом протока како би спречио да брзи предајник претрпа порукама спорог пријемника које он не може да обради.

UDP је једноставан, непоуздан, бесконекциони транспортни протокол за апликације које не захтевају строгу контролу грешака и редоследа пристизања пакета. Ради се о апликацијама као што су оне које преносе аудио и видео, код којих је брза испорука пакета важнија од прецизне испоруке [5].

6.3. TCP/IP – Апликациони слој

TCP/IP модел не предвиђа презентациони и слој сесије, већ су функције ових слојева припојене апликационом слоју. То значи да апликације морају самостално да реализују функције које се односе на сесију и презентацију података, ако су им такве функције уопште потребне.

Апликациони слој садржи већи број протокола високог нивоа. Првобитно су развијени протоколи: *TELNET* (виртуелни терминал), *FTP* (*File TRansver Protocol*) - протокол за пренос фајлова и *SMTP* (*Simple Mail Transfer Protocol*) - протокол за пренос електронске поште.

Временом, апликациони слој је проширен бројним протоколима, од којих су најзначајнији: *DNS* (*Domain Name System*) - за пресликавање имена хостова у њихове мрежне адресе и *XTTP* - за прибављање страна на Web-у.

7. Уређаји између мрежа унутар референтног модела

7.1. Рутер (*Router*)

Да бисте спречили (или бар умањило) лутање пакета кроз мрежу и смањили вероватноћа реемитовања , у глобалним мрежама користе се уређаји названи рутери или усмеравачи. То је мрежни уређај који прослеђује пакете података између рачунарских мрежа унутра *OSI* Слојева.

Рутери су уређаји који помажу у одређивању најбољег начина на који пакет може да стигне до свог одредишта, у складу са стањем саобраћаја кроз мрежу.

Пример: У глобалним рачунарским мрежама није реткост да порука послана из Немачке у Србију на путу до одредишта прође кроз Берлин, Каиро, Јоханесбург, Токио, Тузлу, Копенхаген и Лондон пре него што стигне до Србије (све се ово одвија у свега неколико секунди).

Стога, рутер може изабрати неки, на први поглед потпуно нелогичан, заобилази пут уместо неког логичнијег пута уколико установи да је директни пут оптерећенији протоком података у односу на заобилазни пут.

Слика 32. *Кабловски рутер.*

Слика 33. *Wifi - бежични рутер.*

Рутер се конфигурише и одржава своје табеле рутирања на основу мрежних адреса. Када прими пакет рутер, прво провери да ли је адреса одредишта на истој мрежи као и адреса извора. Ако јесте, пакет се одбацује. У супротном, роутер прослеђује пакет одредишном уређају ако је његова мрежа повезана на роутер или следећем роутеру на путањи до жељеног уређаја.

7.2. ХАБ (*HUB*)

Хаб је мрежни уређај који такође функционише на првом *OSI* слоју (физичком слоју). На *ХАВ*-у постоји више конектора (обично су то RJ – 45 конектори). На сваки конектор се прикључује по један кабал, преко којег се повезује по једна радна станица или сервер.

Slika 34. *Мрежни уређај ХАВ.*

Slika 35. *RJ – 45 конектор.*

Hub као уређај полако нестаје из рачунарских мрежа због све ниже цене switch уређаја који нуде знатно боље перформанце.

7.3. СВИЧ (*SWITCH*) – Скретница

То је уређај који прослеђује податке од једног мрежног сегмента до другог путем одређене линије. На себи има већи број портова. За разлику од hub уређаја, switch податке не шаље свим сегментима мреже него само сегменту коме су они упућени. Веома значајна могућност коју switch посједује је да се на сваки порт switch-а може прикључити станица са одговарајућим портом. Најчешћи проблем који се јавља код употребе switch-а је преоптерећење.

Slika 36. *Switch омогућава поделу LAN[4] – а на више сегмента.*

[4] Локална рачунарска мрежа или локална мреже (*local area network*, *LAN*) је дизајниран за повезивање рачунара и друге мрежне уређаје на краћим дистанцама, нпр у оквиру једне пословне зграде, постројења или кући.

ЗАКЉУЧАК

Ране осамдесете године прошлог века донеле су огромно повећање у броју и величини мрежа. Како су са временом велике компаније схватале да умрежавањем повећавају добит, а смањују трошкове, додавале су се нове мреже, а повећавале постојеће. То су чинили готово једнаком брзином, којом су се избацивали нови производи и нове мрежне технологије.

Све је то у стопу пратио и технолошки развој мрежа. Средином осамдесетих година почињу потешкоће због извршених проширења. Постајало је све теже и теже организовати мреже да међусобно комуницирају због великог броја различитих мрежних технологија од којих већина није била компатибилна једна са другом. Проблеми који су настали су утицали и на тренутно управљање мрежом и на перспективу даљег развоја. Ово је довело до велике кризе и до хитне потребе за стандардизацијом и организацијом мрежа као низ слојева или нивоа. Број слојева, као и њихова имена разликују се од мреже до мреже. У свим мрежама намена сваког слоја је да пружи одређену услугу вишим слојевима.

Из већ наведених чињеница закључујемо да су мрежни протоколи веома важани за рачунарске комуникације. Јасно је да је, у оволиком мноштву рачунара, неопходно успоставити неки систем идентификације појединих рачунара у мрежи и неки ред по којем ће се све то одвијати. Зато постоји сет правила које смо већ дефинисали као протокол који олакшава комуникацију између рачунара.

За неке од многобројних протокола, можемо рећи да они међусобно сарађују. То значи да у мрежи мора да ради више протокола заједно. Рад више протокола мора да буде усаглашен како се не би догађали конфликти или некомплетне операције, односно некомплетан пренос информација. Као резултат тог усаглашавања настаје слојевитост протокола.

На крају треба споменути и мрежну опрему која је такође важан сегмент у преносу наше поруке. Ти уређаји, као што су рутери, хаб-ови и свичеви-еви, нам гарантују најбољи, најбржи и најефикаснији пут преноса информација.

ЛИТЕРАТУРА

[1] **Стојановић Д.** (2005) Увод у рачунарство, Универзитет у Нишу, Електронски факултет, Ниш.

[2] **Јурић Жељко** (2003) Информатика ISBN 9958-21-208-0, Сарајево Publishing, Сарајево.

[3] **Веиновић М. и Јевремовић А**. (2007) Увод у рачунарске мреже, Универзитет Сингидунум Факултет за пословну информатику, Београд.

[4] **William Stallings.** (1996) Computer Networking with Internet Protocols and Technology.

[5] **Douglas E. Comer, David L. Stevens.** (2000) Internetworking with TCP/IP, Client-Server Programming and Applications.

[6] http://www.znanje.org/abc/tutorials/internet_abc/img/i_13033.jpg *(приступљено: јун, 2013.)*

[7] http://sistemac-arhiva.srce.hr/uploads/RTEmagicC_prva.gif.gif*(приступљено: јун, 2013.)*