Permbajtja

1. Hyrje në rrjetat kompijuterike
2. Qëllimi
   1. Vështirësitë që studentët kanë mbi rrjetat kompijuterike
   2. Zgjidhjet e mundshme
3. Zgjidhja me një aplikacion mobile
   1. Zgjidhja nëpërmjet një aplikacioni android
   2. Zgjedhja e platformës android
4. Zhvillimi aplikacionit
   1. Përdorimi i gjuhes “Processing”
   2. Të dhëna mbi programin
   3. Praktikat e mira të programimit
5. Zhvillime të mëtejshme të aplikacionit

RRJETAT KOMPJUTERIKE

Hyrje ne rrjetat kompjuterike

**Prezantimi i shkurter**

Një kuptim bazë të rrjeteve është e rëndësishme për të gjithë kanë menaxhimin e një serveri. Jo vetëm që është e domosdoshme për marrjen e shërbimeve tuaja në internet në mënyrë të rrjedhshme, ajo gjithashtu ju jep pasqyrë për të diagnostikuar problemet. Gjithashtu është e këshillueshme të kenë njohuri bazë edhe persona të cilët merren me teknologji të tijera komunikimi si zhvilluesit në web apo të software-ve të ndryshme, por edhe përdoruesit e thjeshtë siç mund të jenë konsumatorët e telefonisë mobile.

Ky dokument do të japë një pasqyrë bazë të disa koncepteve të përbashkëta të rrjeteve. Ne do të diskutojmë terminologjinë bazë, protokollet e përbashkëta, si dhe përgjegjësitë dhe karakteristikat e shtresave të ndryshme të rrjeteve.

Ky udhëzues është duke operuar në një sistem të lëvisshëm, por duhet të jetë shumë e dobishme kur zbatoni karakteristikat dhe shërbimet që shfrytëzojë rrjetet në serverin tuaj.

Fjalor i shkurtër i network-ut

Para se të fillojmë diskutimin e rrjeteve me thellësi, ne duhet të përcaktojë disa terma të zakonshme që ju do të shihni të gjithë këtë shkrim, por dhe në guidat dhe dokumentacionet e tjera në lidhje me krijimin e rrjeteve.

Këto terma do të zgjerohec në seksionet përkatëse në vijim:

* **Connection (“Kontakti”)**: Në krijimin e rrjeteve, një lidhje ka të bëjë me pjesë të informacionit përkatës që janë transferuar përmes një rrjeti. Kjo në përgjithësi tregon se një lidhje është ndërtuar para transferimit të të dhënave (duke ndjekur procedurat e paraqitura në një protokoll) dhe më pas prishet në fund të transferimit të të dhënave.
* **Paketë** : Një paketë është, në përgjithësi, njësia më themelore që transferohet në një rrjet. Kur komunikimi mbi një rrjet, pakot janë zarfet që bartin të dhënat tuaja (në pjesë) nga një pikë fundore tek tjera.

Pakot apo paketat kanë një pjesë “header” apo kokë që përmban informacion në lidhje me pakno duke përfshirë burimit dhe destinacionin, matësit e kohës, hopet e rrjetit, etj Pjesa kryesore e një pakete përmban të dhënat aktuale që janë duke u transferuar. Ai është nganjëherë i quajtur trupi ose ngarkesa.

* **Network Interface** (ndërfaqe rrjetti): Një ndërfaqe rrjeti mund t'i referohet çdo lloj ndërfaqe software për të rrjeteve hardware. Për shembull, në qoftë se ju keni dy kartat e rrjetit në kompjuterin tuaj, ju mund të kontrolloni dhe të konfiguroni çdo ndërfaqe rrjeti të lidhur me to në mënyrë individuale.

Një ndërfaqe rrjeti mund të jetë e lidhur me një pajisje fizike, ose ajo mund të jetë një përfaqësim i një ndërfaqe virtuale. Pajisja "loopback" (kthyese), e cila është një ndërfaqe virtuale në makinë lokal, është një shembull i kësaj.

* **LAN** : LAN qëndron për "local area network" ("rrjeti zonës lokale"). Ai i referohet një rrjeti ose një pjesë të një rrjeti që nuk është publikisht të aksesueshme në internet  Një shtëpi apo rrjeti zyrës është një shembull i një LAN.
* **WAN** : WAN qëndron për "wide area network" (“rrjeti zonës së gjërë”). Kjo do të thotë një rrjet që është shumë më e gjerë se një LAN. Ndërsa WAN është termi i përshtatshëm për t'u përdorur për të përshkruar rrjetet e mëdha shpërndara në përgjithësi, ai është menduar zakonisht të tregojë internetin, si një i tërë.

Nëse një ndërfaqe është e thënë të jetë i lidhur me WAN, është përgjithësisht supozohet se ajo është e arritshme nëpërmjet internetit.

* **Protokolli** : Një protokoll është një grup i rregullave dhe standardeve që në thelb përcaktojnë një gjuhë që pajisjet mund të përdorin për të komunikuar. Ka një numër të madh të protokolleve në përdorim të gjerë në krijimin e rrjeteve, dhe ato zbatohen shpesh në shtresa të ndryshme.

Disa protokolle të nivelit të ulët janë TCP, UDP, IP, dhe ICMP. Disa shembuj të njohur të protokolleve shtresa e aplikimit, e ndërtuar mbi këto protokolle të ulëta, janë HTTP (për të hyrë në përmbajtjen e internetit), SSH, TLS / SSL, dhe FTP.

* **Port** : Një portë është një adresë në një makinë të vetme që mund të lidhet me një pjesë të veçantë të programeve. Kjo nuk është një ndërfaqe fizike apo vendndodhe, por ajo lejon serverin juaj të jetë në gjendje të komunikojnë duke përdorur më shumë se një aplikikacion.
* **Firewall (Muri i zjarrit)** Një firewall është një program që vendos nëse trafiku që vjen në një server ose shkon jashtë duhet të lejohet. Një firewall zakonisht punon duke krijuar rregullat për cilin lloj të trafikut është e pranueshme dhe në të cilën portë. Në përgjithësi, Firewalls bllokon portat që nuk janë përdorur nga një aplikacion i veçantë në një server.
* **NAT** : NAT qëndron për “network address translation” (“përkthimi adresave të rrejtit”). Kjo është një mënyrë për të përkthyer kërkesat që janë në hyrje të një serveri rutimi apo rrugëzimi me pajisjet përkatëse ose serverat që dinë në lidhje në LAN-in. Kjo zbatohet zakonisht në LAN-et fizike si një mënyrë për të rrugëzuar me anë të një adrese IP në serverat e nevojshme fundorë.
* **VPN** : VPN qëndron për rrjet privat virtual (“virtual private network”). Ajo është një mjet i lidhjes së LAN-eve veçanta përmes internetit, duke ruajtur privatësinë. Kjo është përdorur si një mjet për lidhjen e sistemeve të largët sikur të ishin në një rrjet lokal, shpesh për arsye sigurie.

Ka shumë terma të tjera që ju mund të vijnë, dhe kjo listë nuk mund të përballojë të gjithë termat e mundshëm. Ne do të shpjegojmë termat e tjera që kemi nevojë për ta. Në këtë pikë, ju duhet të kuptoni disa themelore, që do të na mundësojnë për të diskutuar më mirë temat që do të vijnë.

Shtresat e rrjetit

Ndërsa networking është diskutuar shpesh në kushte e topologji në mënyrë horizontale, mes përdoruesve, zbatimi i tij është shtrirë në një mënyrë vertikale përgjatë një kompjuteri ose rrjeti.

Kjo do të thotë se ka teknologjive të shumta dhe protokollet që janë ndërtuar mbi njëri-tjetrin, në mënyrë që komuniki të funksionojë më lehtë. Çdo shtresë e duke u ngjitur më e lartë nxjerr të dhënat e papërpunuara pak më shumë, dhe e bën më të lehtë për t'u përdorur për aplikiacionet dhe përdoruesit.

Ajo gjithashtu ju lejon të rregulloni shtresat më të ulëta në mënyra të reja, pa pasur nevojë për të investuar kohë dhe energji për të zhvilluar protokollet dhe aplikacionet që merren me këto lloje të trafikut.

Gjuha që ne përdorim për të folur për secilin prej skemave të shtresave ndryshon në mënyrë të konsiderueshme në varësi cili model ju përdorni. Pavarësisht modelit të përdorur për të diskutuar shtresat, rruga e të dhënave është e njëjtë.

Si të dhënaat janë dërguar nga një makinë, ajo fillon në krye të stakut dhe filtra poshtë. Në nivelin më të ulët zë vend transmetimi aktual në një tjetër makinë. Në këtë pikë, të dhënat udhëtojnë mbrapa deri nëpër shtresat e kompjuterit tjetër.

Çdo shtresë ka aftësinë për të shtuar të dhënat e saja rreth të dhënave që ajo merr nga shtresa e ngjitur, e cila do të ndihmojë shtresat që vijnë pas të vendosë se çfarë të bëjë me të dhënat, kur ajo është kaluar.

**Modeli OSI**

Historikisht, një metodë e folur në lidhje me shtresat e ndryshme të komunikimit të rrjetit është modeli OSI. OSI qëndron për “Open Systems Interconnect” ose ndërkontaktimi i Sistemeve të Hapura.

Ky model përcakton shtatë shtresa të veçanta. Shtresat në këtë model janë:

* **Aplikimi** : Shtresa e aplikimit është shtresa që përdoruesit dhe aplikacionet bashkëveprojnë më shpesh. Komunikimi në rrjet është diskutuar në aspektin e disponueshmërisë së burimeve, partnerët për të komunikuar me të, dhe sinkronizimi i të dhënave.
* **Prezantimi** : Shtresa prezantimi është përgjegjëse për të drejtuar burimet dhe krijimin e kontekstit. Ajo është përdorur për të përkthyer në nivel më të ulët të dhënat e rrjeteve sesa të dhënat që aplikacionet presin për të parë.
* **Sesioni** : Shtresa sesionit është një mbajtës lidhjeje. Ajo krijon, mirëmban dhe shkatërron lidhjet ndërmjet nyje në mënyrë të vazhdueshme.
* **Transport** : Shtresa e transportit është përgjegjëse për dorëzimin e shtresave mbi të në një lidhje të besueshme. Në këtë kontekst, të besueshme i referohet aftësisë për të verifikuar se një pjesë e të dhënave u prit paprekur apo pu u dëmtuar në anën tjetër të lidhjes.

Kjo shtresë mund të ridërgojë informacion që është rënë ose dëmtuar dhe mund të konfirmojë marrjen e të dhënave në kompjuterët e largët.

* **Network**: Shtresa e rrjetit apo networkut është përdorur për të rrugëzuar tëdhënat mes nyjeve të ndryshme në rrjet. Ajo përdor adresa që të jetë në gjendje për të të treguar se cilin kompjuter duhet për të dërguar informacion. Kjo shtresë mund të ndajë mesazhe të mëdha në copa të vogla të ribashkueshme në anën e kundërt.
* **Data Link** : Kjo shtresë është zbatuar si një metodë për krijimin dhe ruajtjen e lidhjeve të besueshme në mes të nyjeve të ndryshme ose pajisjeve në një rrjet duke përdorur lidhjet ekzistuese fizike.
* **Fizike** : Shtresa fizike është përgjegjëse për trajtimin e pajisjeve aktuale fizike që janë përdorur për të bërë një lidhje. Kjo shtresë përfshin softwaren që menaxhon lidhjet fizike, si dhe vetë hardware-t (si Ethernet).

Siç mund ta shikoni, ka shumë shtresa të ndryshme që mund të diskutohen në bazë të afërsisë së tyre me hardware dhe funksionalitetin që ato ofrojnë.

**Modeli TCP / IP**

Modeli TCP / IP, zakonisht më i njohur si “protokoll Internet suite”, është një model tjetër me shtresa por është më e thjeshtë dhe është miratuar gjerësisht. Ai përcakton katër shtresa të veçanta, disa prej të cilave përputhen me modelin OSI:

* **Aplikimi**: Në këtë model, shtresa e aplikimit është përgjegjës për krijimin dhe transmetimin e të dhënave të përdoruesit mes aplikacioneve. Aplikacionet mund të jetë në sisteme të largëta, dhe duhet të paraqiten si të vepronin si të ishin lokalish tek përdoruesit.

Komunikimi thuhet të zhvillohet në mes të partnerëve.

* **Transport**: Shtresa e transportit është përgjegjëse për komunikimin mes proceseve. Ky nivel i rrjeteve përdor portat për të trajtuar shërbime të ndryshme. Ai mund të ndërtojë lidhje të besueshme ose të pabesueshme në varësi të llojit të protokollit të përdorura.
* **Internet** : Shtresa internet është përdorur për të transportuar të dhëna nga nyja në nyje në një rrjet.Kjo shtresë është e vetëdijshëme për pikat fundore të lidhjeve, por nuk shqetësohet për lidhjen aktuale të nevojshme për të marrë nga një vend në tjetrin. IP adresat janë të përcaktuara në këtë shtresë si një mënyrë për të arritur sistemet e largëta në një mënyrë të adresueshme.
* **Link** : Shtresa e Lidhjes zbaton topologjinë aktuale e rrjetit lokal që lejon shtresa internet për të paraqitur një ndërfaqe adresueshme. Ajo krijon lidhjet midis nyjeve fqinje për të dërguar të dhëna.

Siç mund ta shikoni, modeli TCP / IP, është pak më abstrakt dhe fluid. Kjo e bëri më të lehtë për t’u zbatuar dhe për të lejuar që ajo të bëhet mënyra dominuese sesi rrjetat e shtresëzuara janë të kategorizuar.

Ndërfaqet

Interfaces apo ndërfaqet janë pikat e komunikimit të rrjeteve për kompjuterin tuaj. Çdo ndërfaqe është e lidhur me një pajisje të rrjeteve fizike apo virtuale.

Në mënyrë tipike, serveri juaj do të ketë një ndërfaqe konfigurimi të rrjetit për çdo Ethernet ose kartë interneti pa tel që ju keni.

Përveç kësaj, ajo do të përcaktojë një ndërfaqe rrjeti virtual të quajtur "loopback" ose ndërfaqe localhost pra e makinës lokale. Kjo është përdorur si një ndërfaqe për të lidhur aplikimet dhe proceset në një kompjuter të vetëm për aplikimet dhe proceset e tjera. Ju mund të shihni këtë referohet si interface "lo" në shumë mjete.

Shumë herë, administratorët konfigurojnë një ndërfaqe të trafikut të shërbimeve në internet dhe një tjetër ndërfaqe për një LAN apo rrjet privat.

Protokollet

Networku punon nga mbajtja siper e një numëri protokollesh të ndryshëm në krye të njëri-tjetrit. Në këtë mënyrë, një pjesë e të dhënave mund të transmetohet duke përdorur protokolle të shumta të enkapsuluara tek njëri-tjetrin.

Ne do të flasim për disa nga protokollet më të zakonshme që ju mund të shihni dhe të përpiqemi për të shpjeguar dallimin, si dhe të japim kontekst se çfarë pjesë e procesit që janë të përfshirë.

Ne do të fillojë me protokollet e zbatuara në shtresat e ulëta të rrjeteve dhe të vazhdojmë rrugën tonë deri në protokollet me abstraksion të lartë.

**Protokolli “Media Access Control”**

Media access control është një protokoll i komunikimit që është përdorur për të dalluar pajisje të veçanta. Çdo pajisje është menduar për të marrë një adresë unike MAC gjatë procesit të prodhimit që e dallon atë nga çdo pajisje të tjera në internet.

Duke iu drejtuar hardware me adresën MAC ju lejon referencës së një pajisjeje me një vlerë të veçantë edhe kur software-t në krye mund të ndryshojë emrin për atë pajisje të veçantë gjatë operacionit.

MAC është një nga protokollet e vetme nga shtresa e Lidhjes apo Data Link-ut që kanë gjasa për të bashkëvepruar me mbi një bazë të rregullt.

Në këtë shtresë mbi këtë MAC ngrihen disa protokolle për përdorimin e kanalit të komunikimit. Të tillë kemi ALOHA një protokoll i cili lejon çdo host të dërgojë pa u kujdesur fare nëse kanali është i zënë, kjo e bën këtë protokoll të dobët pasi ka përplasje të shumta të paketave. Kemi dhe protokolle të tjere të cilët pranojnë përplasjet dhe të tjerë që kujdesen shumë për momentin që do të dërgojnë duke dëgjuar nëse kanali është bosh.

**IP**

IP protokoll është një nga protokollet themelore që lejojnë internet për të punuar. IP adresat janë unike në çdo rrjet dhe ata lejojnë makina për të kuptuar njëra-tjetrën nëpër një rrjet. Ai zbatohet në shtresën e internetit në modelin IP / TCP.

Rrjetet mund të jenë të lidhura së bashku, por trafiku duhet të rrugëzohest kur kalojnë kufijtë e rrjetit. Ky protokoll supozon një rrjet të besueshëm dhe shtigje të shumta për të njëjtin destinacion që mund të ndryshojë në mënyrë dinamike.

Ka një numër të ndryshëm të zbatimeve protokollit. Zbatimi më i zakonshëm sot është IPv4, megjithatë IPv6 edhe pse është në rritje si një alternativë për shkak të mungesës së adresave në dispozicion të IPV4 dhe përmirësimeve tek protokollet.

IPV4 funksionon me një numër binar me 32 bit, por që shpesh herë shprehet dhe me 4 numra treshifror dhjetorë.

**TCP**

TCP qëndron për protokoll i kontrollit të transmetimit “transmission control protocol”. Ai zbatohet në shtresën e transportit të IP / modelit TCP dhe është përdorur për të krijuar lidhje të besueshme.

TCP është një nga protokollet që enkapsulon të dhënat në pako. Ai pastaj transferon ato në fund e largët të lidhjes duke përdorur metodat në dispozicion në shtresat më të ulëta. Në anën tjetër, ajo mund të kontrolloi për gabime, të kërkojnë pjesë të caktuara të ridërgohen, dhe ringjisë informacionin në një copë logjike për t’a dërguar në shtresën e aplikimit.

Protokolli ndërton një lidhje para se transferimit të të dhënave duke përdorur një sistem të quajtur një shtrëngim duarsh tre-palësh. Kjo është një mënyrë për të dy skajet e komunikimit të pranojë kërkesën dhe të bien dakord mbi një metodë për të siguruar besueshmërinë e të dhënave.

Pasi janë dërguar të dhënat, lidhja është nisur poshtë duke përdorur një “shtrëngim duarsh” katër-palësh të ngjashëm.

TCP është protokolli i zgjedhur për shumë nga përdorimet më të popullarizuara për internet, duke përfshirë WWW, FTP, SSH, dhe email. Është e sigurt të themi se interneti që ne njohim sot nuk do të ishte këtu pa TCP.

**UDP**

UDP qëndron për Protocol User Datagram (dërgesat e përdoruesit). Ky është një protokoll shoqërues për TCP dhe zbatohet edhe në shtresën e transportit.

Dallimi themelor në mes UDP dhe TCP është se UDP ofron transferimin jo të besueshme të dhënave. Kjo nuk do të verifikojë që të dhënat janë marrë në anën tjetër të lidhjes. Kjo mund të tingëllojë si një gjë e keqe, dhe për shumë qëllime, ajo është e tillë. Megjithatë, është gjithashtu shumë e rëndësishme për disa funksione.

Për shkak se ajo nuk është e nevojshme të lejë të presim për konfirmimin nëse të dhënat janë marrë dhe të detyrohemi për të ridërgim të të dhënave, UDP është shumë më e shpejtë se TCP. Ajo nuk ka krijuar një lidhje me pritës të largët, ajo thjesht hedh të dhënat dhe nuk e kujdesit të presë në qoftë se është e pranuar apo jo.

Për shkak se ajo është një transaksion i thjeshtë, ajo është e dobishme për komunikim të thjeshtë si “querying” për burimet e rrjetit. Ajo gjithashtu nuk mban një gjendje, që e bën të madhe për transmetimin e të dhënave nga një makinë në shumë klientë në kohë reale. Kjo e bën atë ideal për VOIP, lojra, dhe aplikacionet e tjera që nuk mund të përballojë vonesa.

**HTTP**

HTTP qëndron për protokollin e transferimit të hipertekstit (transmission control protocol). Ky është një protokoll përcaktuar në shtresën e aplikimit që formon bazën për komunikim në web.

HTTP përcakton një numër të funksioneve që tregojnë sistemin e largët që ju jeni duke kërkuar. Për shembull, GET, POST, dhe DELETE të gjitha të ndërveprojnë me të dhënat e kërkuara në një mënyrë të ndryshme. Ai është protokolli më i përdorur sot.

**FTP**

FTP qëndron për protokoll i transferimit të file-ve “file transfer protocol”. Ai është gjithashtu në shtresën e aplikimit dhe siguron një mënyrë për të transferimin e skedarëve të plotë nga një host në një tjetër.

Kjo është e natyrshisht e pasigurt, kështu që nuk është e rekomanduar për ndonjë rrjet vetëm nëse është zbatuar si publik, thjesht për shkarkim.

**DNS**

DNS qëndron për “domain name system”. Kjo është një protokoll i shtresës së aplikimit i përdorur për të siguruar një mekanizëm emërtimin e lexueshëm nga njerëzit për burimet e internetit. Kjo është ajo që lidh një emër domain për një adresë IP dhe ju lejon për të hyrë në faqet me emër në shfletuesin apo browser-in tuaj.

**SSH**

SSH qëndron për “secure shell” pra një zgavër apo guackë e sigurtë. Ky është një protokoll i koduar zbatuar në shtresën e aplikimit që mund të përdoret për të komunikuar me një server në distancë në mënyrë të sigurt. Shumë teknologji të tjera janë ndërtuar rreth këtij protokolli për shkak të këtij enkriptimi në çdo anë.

Ka shumë protokolle të tjera që ne nuk i kemi mbuluar që janë po aq të rëndësishëm. Megjithatë, kjo duhet të ju jap një pasqyrë të mirë të disa prej teknologjive themelore që bëjnë internetin dhe rrjetet të jenë të mundura.

**Algoritmat e rutimit**

Algoritmat e rutimit ose të rrugëzimit ashtu si thotë dhe emri janë algoritma të përdorura nga pajisjet në rrjet për të dërguar paketat. Ato implementohen mbi ruterat në mënyre që këta të fundit të dërgojë në mënyrë të suksesshme dhe të shpejtë paketat apo informacionet në destinacion. Algoritmat ndahen në algoritma statikë ku fillimisht bëhet planifikimi dhe më pas gjithçka vahzdon të veptojë me të njëjtën metodë dhe ato dinamikë të cilët i adaptohen situatës. Ideja fillestare është hedhja e paketës në ruterin më të afërt, kjo njihet si “shortest-path-first”. Por kjo metodë nuk është efikase shpesh herë pasi nuk mer parasysh faktorët e tjerë. Të tjera algoritma që mund të përmendim janë: “floodin”, e cila e përmby rrjetin me paketa kur nuk e njeh drejtimin se ku ta çojë; “Distance Vector Routing”, e mabn një tabelë të ndryshueshme me të dhënat e distancave nga ruterat e tjerë; “Link state routing”, i i cili është një algoritëm dinamik dhe bën disa matje para se të fillojë të dërgojë, por dhe bën adoptime në sajë të trafikut, etj.

Përfundim

Në këtë pikë, ju duhet të jenë të njohur me disa terminologjinë themelore të rrjeteve dhe të jeni në gjendje për të kuptuar se si komponentët e ndryshme janë në gjendje të komunikojnë me njëri-tjetrin. Në vazhdim po shpjegojmë disa nga pikat të cilat janë janë trajtuar në aplikacionin android “Bazat e network-ut”.

https://www.digitalocean.com/community/tutorials/an-introduction-to-networking-terminology-interfaces-and-protocols

ALOHA

Aloha është një sistem për koordinimin dhe arbitrimin në një kanal të përbashkët në rrjetet e komunikimit. Ajo është zhvilluar në 1970 nga Norman Abramson dhe kolegët e tij në Universitetin e Hawaii. Sistemi origjinal përdorej për transmetime radio, por sistemi është zbatuar në sistemet e komunikimit satelitor.

Një sistem i përbashkët i komunikimit si Aloha kërkon një metodë të trajtimit të përplasjeve që ndodhin kur dy ose më shumë sisteme kërkojnë për të transmetuar në kanal në të njëjtën kohë. Në sistemin aloha, një nyje transmeton kur ka të dhëna në dispozicion për të dërguar.Nëse një nyje tjetër transmeton në të njëjtën kohë, një përplasje ndodh, dhe paketat që janë të transmetueshme humbasin. Megjithatë, një nyje mund të dëgjojë transmetimet, madje edhe të vetat, dhe të përcaktojë nëse paketat janë transmetuar.

**Aloha do të thotë "Hello".** Aloha është ‘’multiple access [protocol](http://ecomputernotes.com/computernetworkingnotes/computer-network/protocol)’’ në shtresën DATALINK dhe propozon si terminale të shumta të hapur të aksesojnë pa ndërhyrje apo përplasje. Në vitin 1972 Robert zhvilloi një protokoll që do të rrisë kapacitetin e aloha dy herë. Protokolli Slotted Aloha përfshin ndarjen e intervali kohor në pjesë diskrete dhe çdo interval “slot” korrespondon me periudhën kohore për një paketë apo grup paketash. Kjo metodë kërkon sinkronizimin midis nyjeve dërguar për të parandaluar përplasjet.

Ka dy versione ndryshme të aloha:

(i) Aloha  të pastër   
(ii) Slotted aloha

**(I) ALOHA e pastër**

**• Në** Aloha pastër, stacionet transmetojnë korniza sa herë që ata kanë të dhëna për të dërguar.

• kur dy ose më shumë stacione transmetojë njëherësh, ka goditje dhe “framet” shkatërrohen.

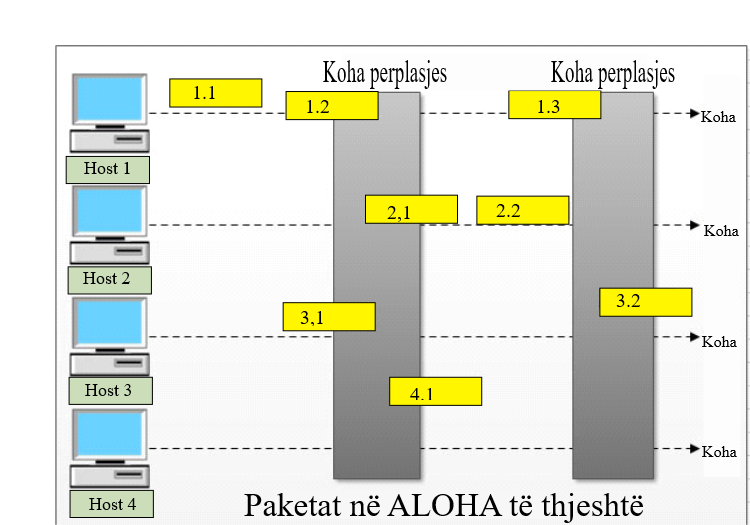
• Në Aloha pastër, sa herë që ndonjë stacion transmeton një “frame”, ajo pret njohjen nga pranuesit.

• Në qoftë se pranimi nuk është marrë brenda afatit të caktuar, stacioni supozon se “fram-a” (ose pranimi) është shkatërruar.

• Nëse “frame” është shkatërruar për shkak të përplasjes stacioni pret për një sasi të rastit të kohës dhe ia dërgon atë përsëri. Këtë herë afati pritjes duhet të jetë i rastësishëm, ndryshe “frame” e njëjta do të përplaset përsëri dhe përsëri.

• Prandaj Aloha e pastër dikton se kur afati kalon, çdo stacion duhet të presë për një çast të rastit para se ridërgojë “frame” e saj. Kjo kohë e rastësishme do të ndihmojë për të shmangur goditjet me shume.

• Figura tregon një shembull të përplasjeve kornizë në Aloha pastër.



• Në fig ka katër stacione që kanë ndarë me njëri-tjetrin për përdorim në kanal të përbashkët. Të gjitha këto stacione janë duke transmetuar “frame”. Disa prej këtyre “frame” përplasen për shkak se “frame” të shumta janë në garë për kanalin përbashkët. Vetëm dy korniza, “frame” 1.1 dhe 2.2 kornizë arrijnë të mbijetojnë. Të gjitha “frame”e tjera janë shkatërruar.

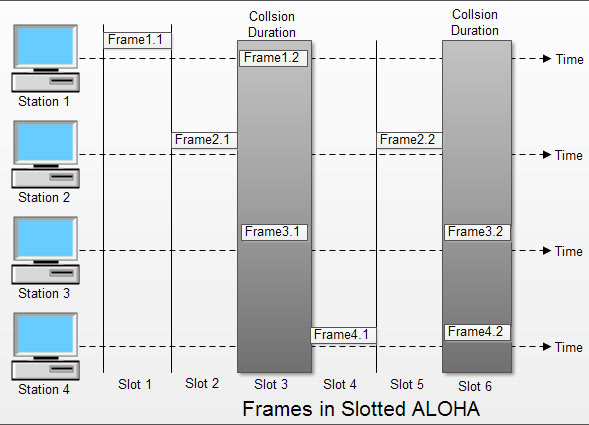
• Sa herë që dy “frame” do të përpiqen për të zënë kanalin në të njëjtën kohë, do të ketë një përplasje dhe të dy do të jetë të dëmtuara. Nëse bit i parë i një “frame” të re përputhet me vetëm pak nga fundit i një “frame” tjetër pothuajse të përfunduar, të dy “frame” do të shkatërrohen krejtësisht dhe të dy do të duhet të ritransmetohen.

**(Ii) Slotted aloha (aloha e fragmentuar në kohë)**

• Slotted Aloha u shpik për të përmirësuar efikasitetin e Alohas së pastër sepse shanset e përplasjes në Aloha pastër janë shumë të larta.

• Në slotted aloha, koha e kanalit të përbashkët është i ndarë në intervale diskrete të quajtura slots.

• Stacionet mund të dërgojë një “frame” vetëm në fillim të slotit dhe vetëm një “frame” dërgohet në çdo slot.



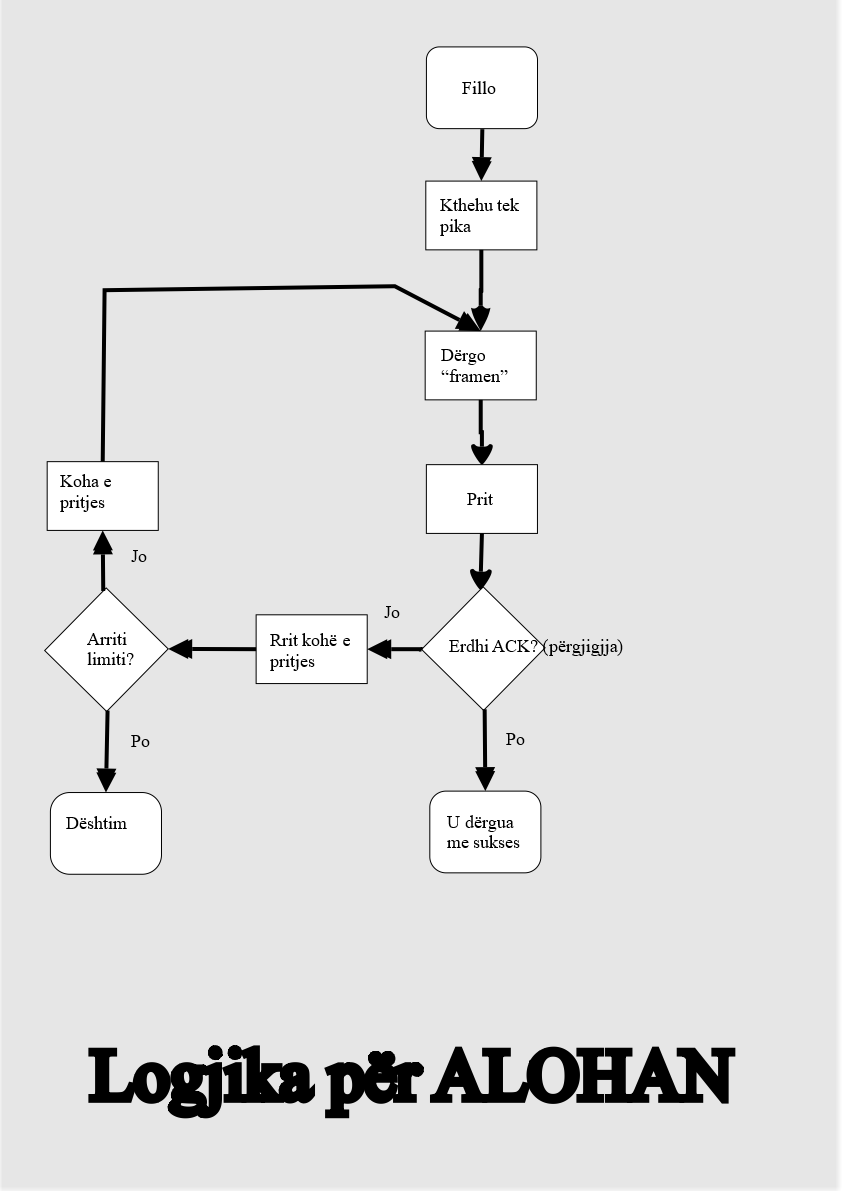
Në slotted aloha, nëse stacioni nuk është në gjendje për të vendosur “frame” mbi kanalin në fillim të slotit  *Psh*  nëse humbet kohën e slotit të vet atëherë stacioni duhet të presë deri në fillim të slotit tjetër.

• Në slotted aloha, ka ende një mundësi e përplasjes në qoftë se dy stacione përpiqen për të dërguar në fillim të të njëjtit segment kohor siç është paraqitur në figurë.

• Slotted Aloha ende ka një avantazh mbi Aloha pastër pasi shanset e përplasjes janë reduktuar në një gjysmë edhe pse transmetimi eshtë më i kufizuar.

## **Protokolli Flow Chart për aloha:**

Figura tregon grafikun e rrjedhës protokoll për Aloha.



**Shpjegim:**

• Një stacion i cili ka një “frame” të gatshme do të dërgojë atë.

• Pastaj ai pret për disa kohë.

• Nëse ai merr njofimin mbapsht, atëherë transmetimi është i suksesshëm.

• Përndryshe stacioni përdor një strategji pritjeje, dhe e dërgon paketën përsëri.

• Pas shumë herëve nëse nuk ka njoftim, atëherë stacioni anullon idenë e transmetimit.

Burim reference: http://ecomputernotes.com/computernetworkingnotes/communication-networks/what-is-aloha

Distance Vector Routing (DVR)

Protokollet eDistance Vector Routing bazojnë vendimet e tyre në rrugën më të mirë për një destinacion të caktuar bazuar në distancë. Distanca matet zakonisht në HOPS, edhe pse distanca metrikë mund të jetë vonesa, pako humbur, apo diçka të ngjashme. Nëse distanca është metrikë hop, atëherë çdo herë që një pako shkon përmes një router, një hop konsiderohet të ketë përshkuar. Rruga me numrin më të vogël të HOPS në një rrjet të caktuar është konstatuar të jetë rruga më e mirë drejt këtij rrjeti.

Vektori tregon drejtimin në atë rrjet të veçantë. Protokollet Distance vektoriale dërgojnë gjithë tabelën e rutimit për fqinjët e lidhur direkt. Shembuj të protokolleve vektor distancë përfshijnë **RIP - Routing Information Protocol** dhe **IGRP - Brendshme Gateway Routing Protocol**.

Nëse jeni të interesuar në gjetjen e më shumë informacion mbi RIP, shikoni shkrimet e mia se si të konfigurosh Routing Information Protocol[RIPv1](https://www.pluralsight.com/blog/cisco-how-to-configure-rip-1) dhe [RIPv2](https://www.pluralsight.com/blog/cisco-how-to-configure-rip-2) .

# Address Resolution Protocol (ARP)

Përkufizimi - Çfarë Address Resolution Protocol (ARP) do të thotë.

Address Resolution Protocol (ARP) është një protokoll rrjeti i nivelit të ulët për përkthimin e adresave nga shtresa e rrjetit në adresat shtresën e link-ut.   
  
ARP shtrihet në mes të shtresave 2 dhe 3 të modelit OSI, edhe pse ARP nuk është përfshirë në kuadrin OSI dhe lejon kompjutera për të futur njëri -tjetrin nëpër një rrjeti përpara komunikimit.   
  
Për shkak se protokollet janë njësi bazë të komunikimit të rrjetit, adresa është e varur në protokollet si ARP, e cila është e vetmja metodë e besueshme e trajtimit detyrat e kërkuara.

Kur konfigurohet një kompjuter të ri rrjeti, çdo sistem është caktuar mbi një adresë Internet Protocol (IP) për identifikimin dhe komunikim primar. Një kompjuter gjithashtu ka një identitet unik me anë të adreses së media access control (MAC).Prodhuesit ja kanë ngjitur adresën MAC në kartën lokale Area Network (LAN). Adresa MAC është e njohur edhe si adresën fizike e kompjuterit.   
  
Para se dy kompjutera të komunikojnë, secili duhet të di IP e tjetrit afërm apo MAC adresat. Nëse kompjuteri A ka vetëm MAC adresën e kompjuterit B, kompjuteri A mund të zbulojë IP adresën e saj duke dërguar një kërkesë ARP për kompjuter B. Kompjuteri B pastaj mund të përgjigjti duke bashkangjitur adresën e tij IP me ARP për kompjuterin A. Kjo adresë përkthimi dhe ky proces shkëmbimi i thjeshtë është roli kryesor i ARP.   
  
Tabelat ARP mund të ruhen për të rritur normat e transmetimit nga mbajtja e adresave të njohur në rrjetin dhe transmetimin e ndonjë MAC ose IP që ndrzshon nga ARP.   
  
Nuk ka asnjë autentifikim nevojshme në këtë nivel, kështu që levizja e IP dhe MAC Adresat është e mundur. Ndonjë software shtesë mund të jetë i nevojshm të drejtoje tabelat e ARP dhe parandalimin e sulmeve dashakeqe të disa përdoruesve.

Qëllimi i këtij projekti

Të lexosh për protokollet e komunikimit kërkon një kohë të gjatë. Dhe më pas të arrish të kuptosh logjikën e tyre sidomos kur algoritmat vijnë e përparojnë bëhet akoma dhe më e vështirë.

Duke marrë shkas nga mundësitë që ne kemi sot falë teknologjisë duhet gjetur diçka më efikase për studimin e algorimave në rrjete. Për këtë ndoshta duhet diçka vizuale dhe interaktive. Sot këto metoda janë duke u përhapur shumë dhe shihen si më eficiente.

Qëllimi i këtij punimi konsiston në gjetjen e kësaj mënyre mësimdhënieie. Kjo do t’i shërbejë kryesisht studentëve të cilët studiojnë kurse në lidhje me rrjetat, por dhe njerëzve të tjerë të cilët ju interson kjo fushë.

Zgjidhja nëpërmjet një aplikacioni

Sot bota ecën drejt zhvillimit të informacionit dhe dixhitalizimit. Sot kemi shumë tekste, imazhe, vidjo por dhe software të ndryshëm të cilët kanë si synim edukimin. Për këtë një zgjidhje e mundshme e problemit të studimit të lehtësuar të rrjetave kompjuterike do të ishte krijimi i një software i cili do të ishte miqësor dhe interaktiv me përdoruesin.

Për këtë kam arritur të ndërtoj një program akoma në hapat e tij të në platformën mobile android i cili bën të mundur të japë disa ilustrime grafike të disa prej protokolleve të rrjetave. Në të ka edhe pjesë shpjegimi me tekst por qëllimi kryesor i tij janë ana vizuale, interaktiviteti dhe simulimet. Deri tani në janë përfshirë këto pika studimi:

* ALOHA
* Slotted ALOHA
* ARP (“Address Resolution Protocol”)
* DVR (“Distance Vector Routing”)
* Makinë llogaritëse me shpjegime për llogaritjen e subneteve

Zgjedhja e platformës android

Natyrisht mund të lindë pyetja pse pikërisht android. Nëse do të shohim grafikun numri i përdoruesve të pajisjeve të vogla si tabletat apo celularët është rritur shumë vitet e fundit. Kjo sepse ato janë të vogla, merren kudo me vete dhe mund të kenë internet në çdo kohë gjë që na lidh me boten fare shpejt.

Ndërkohë që zgjedhja e android midis sistemeve të tjera operative si iOS apo windows është jo për shkak të numrit të përdoruesve por sepse mundësitë e mia personale afronin më tepër me të.

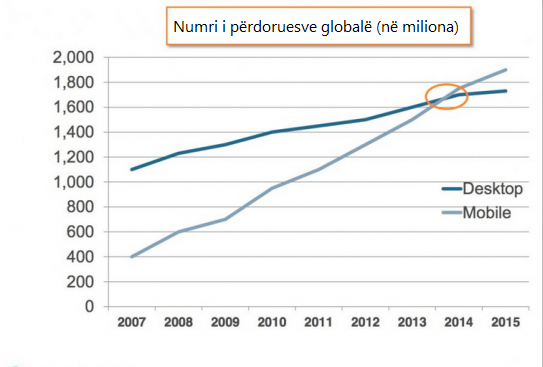


Figura 3.1 Marrë nga comScore, Morgan Stanely Research

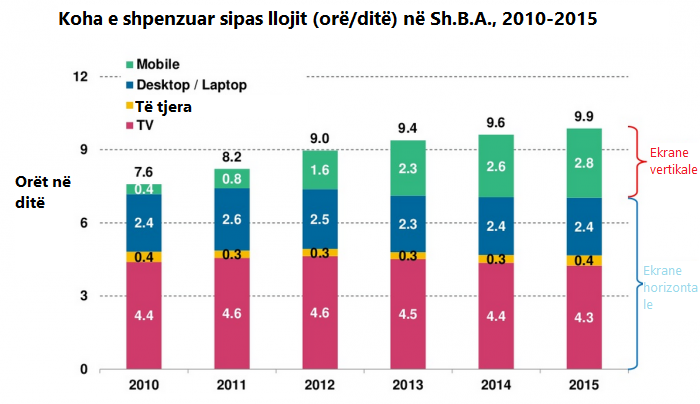


Figura 3.2 Burimi: “eMarker 4/15 Coaute analysis @KPCB”

Zhvillimi aplikacionit

Përdorimi i gjuhes “Processing”

Për krijimin e këtij aplikacioni është përdorur gjuha processing. Gjuha processing është e ngjashme me java-n. Ajo mund të procesojë në mënyrë të shpejtë pjesë grafike. Përveç kësaj zhvilluesit në komunitetin e processing kanë krijuar mbi njëqind librari në lidhje me vizualizimin e të dhënave, compozimin e muzikës, rrjetat, eksportimin e skedarëve 3D dhe programimin e pajisjeve elektronike.

Që nga fillimi, Përpunimi është projektuar si një gjuhë programimi e parë. Ajo u frymëzua nga gjuhët më të hershme, si bazë dhe Logo, si dhe përvojat tona si studentë dhe mësimdhënës arteve pamore kurrikulave themel. Processing është duke zhvilluar aktualisht kryesisht në Boston (në Fathom Information Design), Los Angeles (në UCLA Arts Software Studio), dhe New York City (në NYU’s ITP). Sot përdoret në shumë kurse informatike në botë dhe sipas satistikave studentët të cilët kanë studiuar processing në vt të parë krahasuar me të tjeret që kanë studiuar gjuhë të tjera, kanë një dëshirë 2 herë më të madhe për të vazhduar kurse universitare informatike (Sipas https://www.processing.org/overview Education). Kjo e bën processing një gjuhë shumë motivuese sidomos për të rinjtë në botën e informacionit. Processing aktualish ka 3 versione: ai mund të zhvillohet në java e cila është për desktop, android, javascript, etj. Processing gjithashtu mbulon sistemet operative Windows, Linux dhe OS X, në versione 32 dhe 64 bit.

Të dhëna mbi programin

Programi aktualish mbulon vetëm disa prej çështjeve të rrjetave kompijuterike. Në të kemi demostrim të protokollit ALOHA të thjeshtë, ALOHA-s së fragmentiar ose në slote, algoritmit të rutimit DVR (“distance vector routing”, për më tepër shiko kapitullin e parë), në të përdoruesi mund të simulojë një rrjet me routera dhe të shohë sesi ata llogarisin tabelat e tyre të rutimit. Gjithashtu është menduar për një makinë llogaritëse për llogaritjen e subneteve në IP me klasa si dhe një demostrim i *Address Resolution Protocol*. Në imazhet e mëposhtme kemi disa pamje nga aplikacioni.

Praktikat e mira të programimit

Praktikat e mira te kodimit

Hyrje

Teknikat e mira të kodimit dhe praktikat e programimit janë shenjat dalluese të një programues profesionist. Pjesa më e madhe e programit konsiston në marrjen e një numri të madh të zgjedhjesh të vogla, kështu përpiqesh për të zgjidhur një sërë më të madhe të problemeve. Sa mençurisht këto zgjedhje janë bërë varet në masë të madhe me aftësitë dhe ekspertizën e programuesit.

Ky kapitull trajton disa teknika themelore kodimit dhe ofron një koleksion të praktikave të kodimit nga e cila kemi disa gjëra të cilat janë përdorur edhe në aplikacionin tim. Teknikat e kodimit janë kryesisht ato që përmirësojnë leximin dhe mirëmbajtjen e kodit, ndërsa praktikat e programimit janë kryesisht rritje të performancës.

Lexueshmëri e kodit burim ka një ndikim të drejtpërdrejtë në se sa mirë një zhvillues e kupton një sistem kompjuterik. Mirëmbajtja e kodit i referohet sa lehtë sistemi software mund të ndryshohet për të shtuar veçori të reja, të ndryshuar tiparet ekzistuese, për të rregulluar problemet, ose të përmirësuar punën. Edhe pse lexueshmëri dhe mirëmbajtje janë rezultat i shumë faktorëve, një aspekt të veçantë të zhvillimit të softuerit në të cilin mbi të gjithë zhvilluesve ka ndikim është teknika e kodimit. Metoda më e lehtë për të siguruar që një ekip i zhvilluesve do të japin kodin e cilësisë së duhur është të krijohet një standard kodimi, i cili është i detyrueshëm pastaj në komentet rutinë të kodit.

Standardet e kodimit dhe shqyrtimet e kodit

Një standard i plotë kodimi përfshin të gjitha aspektet e ndërtimit të kodit dhe, ndërsa nga zhvilluesit duhet të ushtrojnë kujdes në zbatimin e tij, ai duhet të ndiqet nga afër. Kodi burimor i plotësuar duhet të pasqyrojnë një stil të harmonizuar, edhe në qoftë se një zhvillues i vetëm ka shkruar kodin në një seancë. Në fillim të një projekti software, të krijojë një standard kodim për të siguruar që të gjithë zhvilluesit për këtë projekt janë duke punuar në bashkëpunim. Kur projekti software do të përfshijnë kod burimi ekzistues, ose kur kryejnë mirëmbajtje mbi një sistem ekzistues software, standardi kodimit duhet të deklarojë si të merren bazat ekzistuese të kodit.

Megjithëse qëllimi kryesor për kryerjen e rishikimeve në kod përgjatë ciklit jetësor të zhvillimit është të identifikojë defektet mbi kod, komentet mund të përdoret gjithashtu për të zbatuar standardet e kodimit në mënyrë uniforme. Aderimi në një standard kodimi mund të jetë i mundshëm vetëm kur ndiqen gjatë gjithë projektit software nga fillimi në përfundimin. Kjo nuk është praktike, as nuk është e kujdesshme, për të vendosur një standard kodues pas faktit të kryer.

Teknikat e kodimit

Teknikat e kodimit përfshijnë shumë aspekte të zhvillimit të softuerit dhe, edhe pse ata zakonisht nuk kanë asnjë ndikim në funksionimin e aplikimit, ato kontribuojnë në një të kuptuar të mirë të kodit burim. Për qëllimet e këtij dokumenti, të gjitha format e kodit burimor janë konsideruar, duke përfshirë programimin, scripting, “Markup languages”, dhe gjuhë “query” ose anketash nëse do t’u shprehnim në shqip.

Teknikat e kodimit të përcaktuara këtu nuk janë propozuar për të formuar një grup të papërkulur të standardeve të kodimit. Përkundrazi, ata kanë për qëllim të shërbejë si një udhërrëfyes për zhvillimin e një standard kodim për një projekt të veçantë software.

Teknikat coding mund t’i ndajmë në tre seksione:

* [Emrat](https://msdn.microsoft.com/en-us/library/aa260844(v=vs.60).aspx#cfr_names)
* [Koment](https://msdn.microsoft.com/en-us/library/aa260844(v=vs.60).aspx#cfr_comments)et
* [Format](https://msdn.microsoft.com/en-us/library/aa260844(v=vs.60).aspx#cfr_format)i

Emrat

Ndoshta një nga ndihmat më me ndikim për të kuptuar rrjedhën logjike të një kërkese është se si quhen elementet e ndryshme të aplikimit. Një emër duhet të tregojë se "çfarë" dhe jo "si." Duke shmangur emrat që ekspozojnë zbatimin themelor, i cili mund të ndryshojë, të ruajnë një shtresë e abstraksionit që thjeshton kompleksitetin. Për shembull, ju mund të përdorni GetNextRouter()(*marrë nga aplikacioni “Bazat e network-ut”*) në vend të GetNextArrayElement().

Një parim i emërtimit është se ka vështirësi në zgjedhjen e një emri të duhur mund të tregojnë se ju keni nevojë për të analizuar më tej ose të përcaktojë qëllimin e një artikulli. Bëni emrat kohë të mjaftueshme për të qenë kuptimplotë, por të shkurtër të mjaftueshme për të shmangur duke u gojor. Programuar, një emër unik shërben vetëm për të dallojnë një objekt nga një tjetër. emrat ekspresive të funksionojë si një ndihmë për lexuesit e njeriut; për këtë arsye, kjo ka kuptim për të siguruar një emër që lexuesi i njeriut mund ta kuptojë. Megjithatë, të jenë të sigurt se emrat e zgjedhura janë në përputhje me rregullat dhe standardet e gjuhës të zbatueshme s.

Në vijim janë rekomanduar emërtimin teknika:

Rutinat

Shmangni emrat e pakapshëm që janë të hapur për interpretim subjektiv, të tilla si Analyze(), ose xxK8 për një ndryshore. Emra të tillë të kontribuojë në dykuptimësi më shumë se abstraksionit.

Në gjuhë objekt-oriented, ai është i tepërt për të përfshirë emra të klasës në emër të pronave të klasës, të tilla si tab.tabFindRow. Në vend të kësaj, përdorni tab.findRow ()(stepNum *marrë nga aplikacioni “Bazat e network-ut”*)..

Përdorni metodën folje-emër për emërtimin rutinat që kryejnë disa operacion në një objekt të caktuar, të tilla si CalculateInvoiceTotal().

Në gjuhë që lejojnë funksion mbingarkesat, të gjitha mbingarkesa duhet të kryejë një funksion të ngjashëm. Për ato gjuhë që nuk lejojnë funksion mbingarkesat, të krijojë një standard emërtimin që lidhet funksione të ngjashme.

Variablat

Është mirë të vendoset një mbaresë dalluese (si mesatare, Sum, Min, Max, Index) në fund të një emri variable kur duket e përshtatshme.

Është mirë të përdoren palët e kundërta në emrat e variablave, si min / max, fillim / fund, dhe hap / mbyll.

Duhe qënë se shumica e emrave janë ndërtuar nga bashkimi i disa fjalëve së bashku, përdorni format të përzier të shkronjave të mëdha dhe të vogla për të lehtësuar leximin e tyre.Përveç kësaj, për të ndihmuar dallimin midis variablave, përdorni modelin Pascal ( ActivateScene() *marrë nga aplikacioni “Bazat e network-ut”*) për emrat e zakonshëm ku shkronjën e parë të çdo fjalë është kapitalizuar. Emrat e variablave, përdorni deve zorrë (stepNum *marrë nga aplikacioni “Bazat e network-ut”*) ku shkronja e parë të çdo fjalë përveç të parës është e kapitalizuar.

Variablat buleane duhet të përmbajë “Is” që nënkupton “PO/JO” ose “True/False”. Psh vlerat të tilla si “IsActive()” (*marrë nga aplikacioni “Bazat e network-ut”*)

Duhet shmangur përdorimi i termave të tilla si Flag/Shenjë/statusi kur kemi emërtimin e variablave të gjendjes, të cilat ndryshojnë nga variablat Boolean se ata mund të kenë më shumë se dy vlera të mundshme. Në vend të documentStatus, përdorin një emër më përshkrues të tilla si documentFormatType.

Edhe për një ndryshore jetëshkurtër që mund të shfaqen në vetëm disa rreshta të kodit, sërisht është mirë të përdoret një emër kuptimplotë. Përdoren emra me një shkronjë të variablave “i” apo ”j”, vetëm në raste të tilla ciklet e shkurtra for.

Nëse përdoret Konvencioni i emërtimit hungarez i Charles Simonyi, ose ndonjë derivat të tij, të zhvillohet një listë e prefikseve standarde për projektin për të ndihmuar zhvilluesit vazhdimisht me emërtimin e variablave.

Për emrat e variablave, ndonjëherë është e dobishme për të përfshirë ndonjë simbol që tregon qëllimin e variablit, të tilla si prefixing një g\_për variablat globale.

Konstantet duhet të jetë mbi të gjitha me shkronja të mëdha me vijë poshtë mes fjalëve, të tilla si NUM\_DAYS\_IN\_WEEK. Gjithashtu, fillojnë grupet e enumeracioneve me një prefiks të përbashkët, si FONT\_ARIAL dhe FONT\_ROMAN.

Të tjera

Duhet të kemi minimizuar përdorimin e shkurtesave. Në qoftë se shkurtimet janë përdorur, të jenë në përputhje me përdorimin e tyre. Një shkurtim duhet të ketë vetëm një kuptim dhe po ashtu, çdo fjalë shkurtuar duhet të ketë vetëm një shkurtim. Për shembull, në qoftë se duke përdorur *min* për të shkurtuar *minimum* , ta bëjmë këtë kudo dhe më vonë nuk e përdorim atë për të shkurtuar “*minutë”* .

Kur emërtimin e funksioneve, të përfshijë një përshkrim të vlerës duke u kthyer, të tilla si getValueLabel()(stepNum *marrë nga aplikacioni “Bazat e network-ut”*).

Emrat e file-ve dhe skedarëve/dosjeve, si emrat e procedurës, duhet të përshkruajnë me saktësi çfarë qëllimi ata shërbejnë.

Shmangni emrat ripërdorimin për elemente të ndryshme, të tilla si një emër rutinë i quajtur ProcessSales()dhe një ndryshore të quajtur iProcessSales.

Shmangni homonyms kur emërtimin elementet për të parandaluar konfuzionin gjatë shqyrtimeve të kodit, të tilla si *“write”* dhe *“right”* .

Kur emërtimin elemente, për të shmangur duke përdorur fjalë që zakonisht keqshqiptohen ose shqiptohen me vështirësi. Gjithashtu, të jenë të vetëdijshëm për dallimet që ekzistojnë mes anglishtes amerikane dhe britanike, të tilla si “color/colour” dhe “check/cheque”. Por dhe fjalë të tjera që shpesh shkruhen gabim. Kjo sepse do të ketë vështirësi në kërkim nëpër kod dhe problemi shtohet edhe më shumë kur ka më shumë se një zhvillues.

Shmangni përdorimin shenjat tipografike për të identifikuar llojet e të dhënave, të tilla si $ për stringat ose% për numrat.

Komentet

Dokumentacioni Software ekziston në dy forma, të jashtëm dhe të brendshëm. Dokumentacioni i jashtëm është ruajtur jashtë kodit burim, të tilla si specifikimet, fotografi ndihmë, si dhe dokumentet e projektimit. Dokumentacioni i brendshëm është i përbërë nga komentet që zhvilluesit shkruani në kodin burim në kohën e zhvillimit.

Një nga sfidat e dokumentacionit software është siguruar se komentet janë të mirëmbajtura dhe të përditësuar paralelisht me kodin burimor. Edhe pse duhet komentuar kodin burim i shërben asnjë qëllimi në kohë të drejtuar, ai është i paçmuar për një zhvillues i cili duhet të mbajë një pjesë veçanërisht të ndërlikuar ose të rëndë të softuerit.

Në vijim janë rekomanduar teknika komentimi:

Kur modifikoni kodin, gjithmonë mbani koment të përditësuar.

Në fillim të çdo rutine pune, ajo është e dobishme për të siguruar standarde, komente të njëllojtë, duke treguar qëllimin e punës, supozimet dhe kufizimet. Një koment unik duhet të jetë një hyrje të shkurtër për të kuptuar pse ekziston kjo punë apo ndërhyrje dhe çfarë mund të bëjë.

Shmangni duke shtuar komente në fund të një linj kodi; Komentet e end-line bëjnë kodin më të vështirë për t’u lexuar. Megjithatë, komentet fund të linjës janë të përshtatshme kur shfaqim deklarata ndryshueshme. Në këtë rast, lidh të gjitha komentet e end-line në një pozicion të përbashkët.

Shmangni përdorimin komenteve rrëmujë, të tilla si një linjë të tërë plot me ylla. Në vend të kësaj, përdorni hapësirën e bardhë për të ndarë komentet nga kodi.

Shmangni rrethimin e një blloku komenti me një kornizë tipografike. Mund të duket tërheqëse, por është e vështirë për të ruajtur.

Para vendosjes, duhet të kemi hequr të gjitha komentet e përkohshme ose pa lidhje për të shmangur konfuzionin përgjatë punës së ardhshme të mirëmbajtjes.

Nëse keni nevojë për komente për të shpjeguar një pjesë komplekse të kodit, shqyrtojeni kodin për të përcaktuar nëse ju duhet për rishkruar atë.Nëse është e mundshme, mos dokumentoni kod të keq, rishkruajeni atë. Edhe pse performanca nuk duhet të sakrifikohet zakonisht për të bërë kodin lehtë për syrin njerëzor, një ekuilibër duhet të mbahet në mes të performancës dhe mirëmbajtjes.

Përdorni fjali të plota kur shkruani komente. Komentet duhet të qartësojë kodin, nuk shtoni dykuptimësi.

Komentoni gjatë kodimit, për shkak se më shumë gjasa nuk do të ketë kohë për të bërë atë më vonë. Gjithashtu, duhet që ju të merrni një shans për të rihapur kodin që ju keni shkruar, atë që është e qartë sot ndoshta nuk do të jetë e qartë gjashtë javë nga tani.

Shmangni përdorimin e komenteve të tepërta ose të papërshtatshme, si vërejtjet humoristike.

Përdorni komente për të shpjeguar qëllimin e kodit. Ata nuk duhet të shërbejë si përkthimet fjalë për fjalë të kodit.

Koment asgjë që nuk është lehtësisht e dukshme në kodin.

Për të parandaluar përsëritjen e problemeve, gjithmonë përdorni komente në problemet e rregulluara dhe punimet rreth kodit, veçanërisht në një mjedis ekipi.

Përdorni komente mbi kodin që përbëhet nga pozicione dhe degë logjike. Këto janë fushat kryesore që do të ndihmojnë lexuesin kur lexoni kodin burim.

Ndani komentet me shenjat e hapjes (apo mbylljes) se komenteve me hapësirë ​​të bardhë. Duke bërë kështu do të bëhet që komentet të dalin të lehtë për të gjetur kur shihet pa patur ngjyra.

Gjatë aplikacionit, të ndërtohen komente duke përdorur një stil uniformë, me shenjat e pikësimit të qëndrueshme dhe strukturën.

Formati

Formatimi bën organizimin logjik të kodit në mënyrë të jashtezakonshëme. Duke marrë kohë për të siguruar që kodi burim është i formatuar në mënyrë konsistente, rrjedha logjike është e dobishme për veten tuaj dhe për zhvilluesit e tjerë të cilët duhet të deshifroj kodin burim.

Në vijim janë rekomanduar teknika formatimit:

Krijimi i një madhësi standarde për një nënpikë, si katër hapësira, dhe e përdorin atë në mënyrë të vazhdueshme.

Përdorni një font “Monospace” (font që i ka shkronjat me gjerësi të njëjtë) kur publikoni versionet “hard” të kodit burim.

Me përjashtim të konstanteve, të cilat janë të shprehura në mënyrën më të mirë në të gjitha karakteret me shkronjë të madhe me vijë poshtë, përdorin rastin e përzier në vend të “underscore” apo vijës poshtë për të bërë emrat më të lehtë për të lexuar.

Vendose kllapat të hapura dhe të mbyllura vertikalisht në një drejtim, të tilla si:

for (i = 0; i <15; i ++)

{

...

}

Ju mund të përdorni një stil pjerrët, ku kllapat hapëse hapur paraqiten në fund të rreshtit dhe kllapat mbyllëse të shfaqet në fillim të rreshtit, të tilla si:

|  |
| --- |
| if(tab.findRow(R.name,"Destinacioni")!=null){ |
| msg = "Routerat jane lidhur me pare\n"; |
| }  (*marrë nga aplikacioni “Bazat e network-ut”*)  Gjithashtu edhe stili kur kemi shume kllapa qe hapen dhe mbyllen njëra pas tjetrës   |  | | --- | | if (dist(mouseX, mouseY, position.x, position.y)<20) {  return true; | | } else { | | return false; | | } | | } | |

(*marrë nga aplikacioni “Bazat e network-ut”*)

Cilado style është zgjedhur, përdorni atë stil të gjithë kodin burim.

Krijimi i një gjatësi maksimale për komentet dhe kodet për të shmangur dritaren që të lëvizi kodin burim dhe për të lejuar për prezantim të pastër të kopjes “hard”.

Përdorin hapësira para dhe pas shumicën e operatorëve kur kjo nuk ndryshon qëllimin e kodit.

Vendos një hapësirë ​​pas çdo presje në listat presje-kufizuar, të tilla si vlerat e vektorëve dhe argumente, kur kjo nuk ndryshon qëllimin e kodit.

Përdorni hapësirë ​​të bardhë për të siguruar të dhëna organizative në kodin burimor. Duke bërë kështu krijon "paragrafët" të kodit, të cilat duhen për të ndihmuar lexuesit në të kuptuarit e segmentimi logjik të softuerit.

Kur një vijë është e thyer në disa linja, e bëjmë të qartë se linja është e paplotë pa vijën e mëposhtme.

Kur është e përshtatshme, për të shmangur vendosjen e më shumë se një deklaratë për rresht. Një përjashtim është një cikël psh: for (i = 0; i < 6; i++).

Praktikat programim

Zhvilluesit me përvojë ndjekin praktika të shumta të programimit ose “rregulla të gishtit”, e cila zakonisht rrjedh nga mësimet e mësuara me mund. Praktikat e listuara më poshtë nuk janë të gjithë-përfshirëse, dhe nuk duhet të përdoren pa u marrë në konsideratë siç duhet. Programuesit veteran devijojnë nga këto praktika me rastin, por jo pa shqyrtim të kujdesshëm të pasojave të mundshme. Duke përdorur praktikën më të mirë programimit në kontekstin e gabuar mund të shkaktojë më shumë dëm sesa dobi.

Për të ruajtur burimet, të jemi selektiv në zgjedhjen e llojit të të dhënave për të siguruar madhësinë e një variable nuk është tepër i madh.

Mbani jetëgjatësinë e variablave sa më të shkurtër të jetë e mundur, kur variablat përfaqësojnë një burim i caktuar, për të cilat mund të ketë “grindje”, të tilla si një lidhje të bazës së të dhënave.

Mbani vendodhjen e variablave sa më të vogla të jetë e mundur për të shmangur konfuzionin dhe për të siguruar mirëmbajtjen. Gjithashtu, ruajmë kodin burimor, potenciali për të thyer pa dashje pjesë të tjera të kodit mund të minimizohet, nëse shtrirja e variablit është e kufizuar.

Përdorni variablat dhe rutinat për një dhe vetëm një qëllim. Përveç kësaj, shmangni krijimin e rutinve me shumë qëllime që kryejnë një sërë funksionesh që nuk kanë lidhje.

Kur shkruani klasa, shmangni përdorimin e variablave publike. Në vend të kësaj, të përdorni procedurat/funksionet për të siguruar një shtresë e enkapsulimi dhe gjithashtu për të lejuar një mundësi për të validuar ndryshimet e vlerës.

Komponentët pa status janë të preferuar, kur shkallëzimi apo performanca janë të rëndësishme. Hartoni komponentët për të pranuar të gjitha vlerat e nevojshme si parametrave të dhëna në vend të duke u mbështetur mbi pronat e objekteve kur thërrisni metoda. Duke bërë kështu eliminon nevojën për të ruajtur objekt gjendjen mes thirrjeve të metodave. Kur është e nevojshme për të ruajtur gjendje, e konsiderojnë duke përdorur metodat alternative, të tilla si ruajtjen e gjendjes në një bazë të dhënash.

Duhet të mos mbahet hapur lidhjet me dhënat kur përdoren kredencialet e përdoruesit specifik. Lidhjet që janë hapur duke përdorur kredenciale të tilla nuk mund të grumbullohen dhe të ripërdoren, duke humbur kështu të mirat e lidhjeve.

Shmangni përdorimin e detyruar të dhënat e konvertimit, të cilat mund të japin rezultate të papritura. Kjo ndodh kur dy ose më shumë variabla të llojeve të ndryshme të të dhënave janë të përfshirë në të njëjtën shprehje. Kur është e nevojshme për të kryer një kastim për një arsye të parëndësishme, duhet të jepet në një koment shoqërues.

Shmangni përdorimin e variablave në fushën e zbatimit.

Përdorni RETURN në procedurat e ruajtura për të ndihmuar programin nëse e di se procedura ka punuar si duhet.

Përdorni teknika detyruese që në fillim kur është e mundur.

Përdorni  Switch në vend të kontrollit të përsëritura të një variable të përbashkët.

Përfundime

Përdorimi i rregullt i teknikave të kodimit dhe praktikave të mira të programimit i cili krijon kod më aftësi apo cilësi të mira luan një rol të rëndësishëm në performancën dhe kualitetin e software-it. Gjithashtu kjo ndihmon në rishikimet e kodit, e cila thjeshtëson punën në rast të problemeve apo zhvillimeve të mëtejshme. Kjo do të sillte kursim të madh në kohë pasi nuk do të duhet të shkruajmë gjithçka nga e para.

Referenca: https://msdn.microsoft.com/en-us/library/aa260844(v=vs.60).aspx

Zhvillime të mëtejshme të aplikacionit

Sigurisht që pikat e studimit të rrjetave kompijuterike janë të shumta. Deri tani në këtë aplikacion kemi mbuluar vetëm një numër të vogël të tyre. Në të ardhmen a afërt do të ishte mirë të kishim mbuluar edhe disa protokolle të tjera si “Link State Routing”, “Flooding”, “Multicast Routing”, etj. Sigurisht dhe në mësimet aktuale mund të bëjmë disa përmirësime në shpjegimet me tekst, ruajtje të konfiguracioneve për shembull.

Janë të mirëpritur të gjithë zhvilluesit e aplikacioneve të bashkohen në këtë projekt, i cili i vjen në ndihmë jo vetëm studentëve të shkencave kompijuterike por dhe kujdo tjetër që është i nteresuar në fushën e informcaionit dhe komunikimit.

Zgjidhja me një aplikacion mobile