##
##################################

Transportni sloj

_____

@Osnovne funkcije transportnog sloja

-Transport aplikativnih podataka

--enkapsulacija i dekapsulacija

-Dve vrste prenosa:

--Byte-stram - vodi racuna o segmentaciji

--Message-stream - prenos i enkapsulacija celokupnih poruka

-Posiljalac:multipleksiranje, primalac:demultipleksiranje

@Port

-ID od 2B koji identifikuje aplikaciju

-IANA - dodeljuje fiksne portove za pojedinacne aplikacije

--Well-known Ports: - serverske aplikacije (0-1023)

--Registred Ports: -klijentske i serverske (1024-49151)

--Private(Dynamic) Ports: (49152-65535)

@Soket

-jednoznacna identifikacija aplikacije na mrezi sadrzi

--IP adresu

--ID transportnog protokola (TCP/UDP)

--Broj porta

@Klijent-Server komunikacija

-Serverske aplikacije

--otvorene za pristup

--soket: unapred poznata IP adresa i poznat UDP/TCP port

-Klijentske aplikacije

--soket: proizvoljna IP adresa, dinamicki dodeljen TCP/UDP port

$Teams,Zoom.. koriste UDP$

@UDP portokol - osnovno

-Connectionless

-Message-stream protokol

-Enkapsulacija, obelezavanje porta i prenos bez uspostavljanja veze

-Nezavistan prenos svakog paketa

-Nepouzdan i brz

@Zaglavlje UDP

-Source port,Destination port,Length,Checksum - sve po 2B

-Length - duzina podataka ukljucujuci i zaglavlje

--8B UDP zaglavlja i 20B IP zaglavlja

-Checksum- obuhvata i Pseudoheder

--Pseudoheder- izvorisna i odredisna IP adresa, identifikacija UDP i duzina UDP paketa

@Primene UDP

-nije bitna pouzdanost

-periodicna komunikacija

--mala varijacija kasnjenja - jitter

-jednostavne aplikacije

-real-time aplikacije(kontinuitet pristizanja poruka)

-kada je potreban broadcast ili multicast

-slozenije funkcije prepustaju se aplikaciji ili Real-time Transport Protocolu - RTP

@TCP protokol

-Connection-oriented

-byte-stream

-pouzdan nezavistan prenos u oba smera

--point-to-point - ne podrzava multicast/broadcast

--full-duplex

@Byte-stream prenos podataka

-Segmentacija - na strani posiljaoca, objedinjavanje na strani primaoca

@TCP zaglavlje

-Source,Destination port,Checksum - po 2B

-HLEN - 4bita - duzina zaglavlja u jedinicama od 4B

-Window Size - ukupan br bajtova koji se mogu poslati pre nego sto se ceka na potvrdu

-Options

@Numeracija bajtova i segmenata

-SEQ - sequence number

-Inicijalna vrednost- slucajno izabran broj u fazi uspostavljanja veze

-Svaki bajt u nizu aplikativnih podataka ima svoj redni broj relativno u odnosuna inicijalni SEQ

-dobijamo identifikaciju segmenata

--odrzavanje redosleda segmenata na prijemu

--pouzdan prenos segmenata - potvrda prijema segmenta

-ACK- Acknowledgment number

--potvrda prijema kontinualnog niza bajtova

--relativni redni broj SEQ sledeceg bajta koji se ocekuje za prijem

--"ovo je pozicija sledeceg bajta za prijem, a svi prethodni su primljeni"

@Kontrolni flegovi

-SYN -inicijalizacija SEQ

-ACK -polje ACK number je validno

-FIN -poslednji segment, zavrsetak konekcije u jednom smeru

-PSH -momentalna predaja segmenta aplikaciji na prijemu, bez baferovanja

-URG -polje Urgent Pointer je validno

-RST -resetovanje konekcije

@Uspostavljanje sesije

-full-duplex-dva nezavisna para SEQ i ACK

-nezavisno uspostavljanje u oba smera

1.KORAK - -->SEQ,SYN ActiveOpen stanje

2.KORAK - <--ACK=SEQ+1,SEQ,SYN,ACK Estabilished stanje

3.KORAK - -->ACK=SEQ+1,SEQ,ACK,DATA --> OSTVAREN TCP SOCKET

@Raskidanje sesije

1.KORAK - -->SEQ,FIN,LAST_DATA

2.KORAK - <--ACK=SEQ+1,ACK - ZATVORENA SESIJA U 1 SMERU OD KLIJENTA KA SERVERU

3.KORAK - <--SEQ,FIN,LAST_DATA

4.KORAK - -->ACK=SEQ+1,ACK - ZATVORENA U OBA SMERA

@Pouzdan prenos - Reliability

-potvrda primljenih podataka SEQ-ACK

-nezavisno u oba smera

-ceka se za svaki poslati segment- timeout

--vreme tajmera > round trip time RTT - varira tokom vremena

-dinamicko odredjivanje RTT

--RTT=a*RTTold+(1-a)RTTnew, Timeout=b*RTT

--retransmisija segmenta za koji je prosao timeout period

@Oporavak od greske

-na osnovu SEQ zna se gde se nalaze podaci

-ACK se salje samo kada pristigne neki segment

-ACK vrednost se odnosi na poslednji bajt u komtinuitetu

--dok se rupa ne popuni salje se ack za ono pre rupe

-Visestruki acknowledgment

--kad se nakon timeouta popuni stici ce ack za sve ono u medjuvremenu

@Rekonstrukcija redosleda segmenata

-razliciti segmenti mogu da stignu razlicitim putanjama

--promena redosleda prijema

--prijemna strana rekonstruise originalni redosledna osnovu SEQ vr primljenih segmenata

@Kontrola toka - Flow control

-mehanizam prozora (Window)

-sprovodi se na strani koja salje podatke

-oznacava ukupan broj bajtova koji se mogu poslati pre nego sto se ceka na potvrdu

-Prozor obuhvata deo aplikativnih podataka koji se salju:

--pre prozora su: poslati podaci tj oni za koje je primljen ACK

--u prozoru su: poslari podaci koji cekaju ACK i podaci koji se mogu poslati

--posle prozora su: podaci koji se ne mogu poslati

-prozor se pomera kada se dobije ACK na segment na pocetku prozora - Sliding Window

-Kada se popuni prozor:

--obustava slanja novih segmenata

--ceka se potvrda prethodnih

-Kontrola se uspostavlja velicinom prozora

--manji prozor --> sporije slanje

--veci prozor --> u kontinuitetu, brze slanje

-Dinamicko uspostavljanje velicine prozora Dynamic Window

--obe strane se dogovore o velicini

--opterecenje prijemne strane/ gubitak paketa --> zahteva smanjenje velicine prozora

---ako nema gresaka prozor se povecava!

@Kontrola zagusenja - TCP Congestion Control

-posiljalac se dinamicki prilagodjava trenutnom opterecenju

-algoritmi:

1)Slow Start

-Advertised window (AW)

--inicijalna vr prozora postavljena pri uspostavljanju veze

--postavlja primalac --> kontrolise brzinu prijema segmenata

-Congestion window (CW)

--da bi se izbeglo zagusenje, posiljaoc postepeno povecava stvarnu velicinu prozora do AW(maksimuma)

--inicijalna vrednost za CW=1 segment

--CW se povecava svakim primljenim ACK za broj do sad primljenih ACK

--Ako se potvrdjuje svaki segment CW se u svakom koraku povecva 2 puta

--Ako nastane timeout --> novi Slow start! TO je znak da prebrzo saljemo podatke

2)Congestion Avoidance

-izbegavanje zagusenja, linearno povecanje prozora umesto ekspoencijalnog

-ssthresh - Slow Start Treshold Size = prag za pocetk linearnog i kraj eksponencijalnog

--Ako ne stigne ACK za neki segment, sstresh se smanjuje na polovnu poslednje vrednosti

---tada je sstresh=CW/2

3)Fast Retransmit

-Dupli (visestruki) ACK

--odredjeni segment je izgubljen(ili je doslo do promene redosleda)

--naredni segmenti su uspesno primljeni

--1 ili 2 dupla ACK - mozda nije izgubljen - mozda je samo doslo do promene redosleda

--3 dupla ACK -segment je izgubljen

---FAST RETRANSMIT - ponovo slanje segmenta - pre isteka timeout intervala

4)Fast recovery

-Kod fast-retransmit 1 segment je unisten, ali najmanje 3 su uspesno stigla

->Nema potrebe za drasticnim usporenjem slanja podataka

-Nastupa fast-recovery nakon fast-retransmita:

--prozor se smanjuje na vrednost sshtresh

--ide se direktno u Congestion Avoidance, bez Slow-start faze!

@Alati za proveru TCP konekcija

-netstat - prikaz otvorenih TCP konekcija na lokalnom uredjaju

-nmap - prikaz otvorenih TCP portova na udaljenom racunaru

$Na koji segment se odnosi pristigli ACK? $ na PRVI segment u TCP prozoru

$Na koji segment se odnosi dupli ACK$ na PRVI segment pre TCP prozora

@TCP	**UDP - razlike**
connection	connectionless
byte-stream	message-stream
pouzdan	nema garancije prenosa
kontrola toka	nema kontrole
zaglavlja 20B	zaglavlja 8B
sporiji	brzi
veb,mejl,posao kad je vazna brzina	periodicna komunikacija,

@QUIC protokol

-sigurnost = TLS - TRANSPORT LAYER SECURITY

--koriste ga aplikacije za sifrovanje saobracaja

--oslanja se na TCP

-QUIC je Goolglov protokol za sigurnu veb komunikaciju

--umesto dosadasjnjeg HTTPS=TCP+TLS, koristi TCP+TLS a prenos preko UDP

-brzo uspostavljanje veze

-mogucnost zadrzavanja QUIC veze i prilikom promene IP adrese klijenta - mobilnost sa 4g/5g

##
####################################

Aplikativni sloj

_____

@Klijent-server komunikacija

-serverske aplikacije

-klijentske aplikacije - vec pricano gore!

-dvosmerna komunikacija izmedju klijentskih i serverskih soketa

--zahtev od klijenta prema serveru

--odgovor servera prema klijentu

@WWW - web servis

-HTTP protokol - prenos poruka sa posebnim tagovima i ugnjezdenom strukturom HTML - TCP PORT 80

--non-persistent (za svaki zahtev nova TCP veza) i persistent konekcija (koristi se i nakon timeouta)

--stateless (ne pamti se stanje aktivnosti klijenta) i stateful modovi za pracenje konekcije

@Proxy servis - Web Cache

-Posredni server za HTTP protokol

--prethodno zahtevane stranice se kesiraju i cuvaju neko vreme

--za ponovljeni zahtev, cak i od drugog korisnika, vracaju se kesiranim podacima

-Prednosti:

--brzi odziv

--veca privatnost korisnika(skriveno za spoljne servere)

--omogucava kontrolu pristupa

@FTP

-protokol prenosa datoteka

-koristi 2 TCP konekcije:

--Kontrolna konekcija - TCP port 21

---7b ascii tekst za zadavanje komandi

---korisnik koristi poseban program, loguje se, zadaje komande...

---konekcija je otvorena dok je korisnik eksplicitno ne zatvori (quite)

---stateful - pamti se stanje aktivnosti klijenta

--Konekcija za podatke - TCP port 20

---jedna konekcija za prenos jedne datoteke

---prenos u oba smera

----klijent-server: put,STOR

----server-klijent: get,RETR

@Email servis

-Slanje elektronske poste

--server posiljalac salje serveru primaocu

--od posiljaoca maticnog servera krece

--SMTP = Simple Mail Transfer Protocol - TCP port 25

-Preuzimanje elektronske poste

--klijent pristupa maticnom serveru i preuzima pristiglu elektronsku postu

-danas: klijentska aplikacija -> maticni server -> server kome je namenjen mejl-> klijenti kome je namenjen mejl da bi ga procitao mora se zakaciti na server (protokoli: POP3,IMAP)

@Udaljeni pristup uredjajima

-TELNET-

--udaljeni pristup tekstualnoj konzoli, TCP port 23

--moze da se proveri da li je funkcionalan neki TCP port na nekom racunaru ovakvom proverom

-SSH- secure shell

--sifrovani udaljeni pristup tekstualnoj konzoli, TCP port 22

-RDP- remote desktop protocol

--udaljeni pristup grafickoj konzoli

@DNS

-Servis za pretvaranje naziva u IP adrese

-Bazicna upotreba: IP adrese su zgodne za masinsko koriscenje ali neprakticne za korisnike

-Potrebno uvesti simbolicka imena za rad korisnika

-->mapiranje simbolickih naziva uredjaja u IP adrese

-mozemo i IP adrese u nazive !

@DNS hijerarhija

-koren stabla - root ("")

-apsolutni naziv domena

--putanja od cvorova do korena stabla

-relativni naziv domena

--poddomen nekog domena

-nazivi racunara (hostova) pripadaju odredjenom domenu i listovi su u stablu

@DNS Struktura

-puno ime domena ili hosta sastoji se od vise delova(lebela,segmenata):

--svaka labela maks 63 karaktera

--maksimalna duzina punog imena 255 karaktera

--case-INsensitive

-Naziv: aaaa.bbb.cc

--cc je Top Level Domain (TLD)

--aaaa,bbb - labele - poddomeni

--aaaa- labela, moze da bude i ime uredjaja i ime poddomena

@TLD- Top level domains

-TLD = poddomeni root domena

-globalni root domen " "-pripada SAD

com. edu. gov. net. org. mil.

-ccTLD - Country Code TLD

--pripadaju pojedinacnim drzavama

@DNS organizacija

-logicka struktura je fizicki organizovana na distribuiran nacin

-Zona - deo stabla

--sadrzi informacije o pripadajucim domenima

--administrativno pripada jednoj celini

--tekstualna datoteka definisana je na jednom serveru (DNS ili NS- name server)

-Delegacija zona

--zona domena definise nazive poddomena

--topologija domena je tehnicki potpuno nezavisna od topologije fizickog povezivanja u mrezi

---uredjaji iz jednog domena mogu da pripadaju razlicitim fizicki odvojenim mrezama

@Princip rada

-Primarni DNS server (za neki domen)

--DNS server na kom je definisana zona za neki domen

--svi podaci za taj domen i definicije poddomena

-Sekundarni DNS server (za neki domen)

--DNS server koji periodicno preuzima zonu od primarnog DNS servera - transfer zone

--Preporuka je da postoji bar jedan sekundarni DNS server za svaki domen

->Autoritativni DNS serveri (za neki domen)

--DNS serveri koji imaju celokupne zone za odredenje domene

---Primarni i sekundarni DNS serveri, ravnopravna uloga

@Razresavanje imena - nalazenje IP adrese za zadato ime

-DNS serveri razresavaju upite klijenata

--uredjaji imaju lokalno podesene DNS servere kojima salju upite

---Windows ima "Preferred" i "Alternate" DNS servere (nije isto sto i primarni i sekundarni!)

-DNS Resolver

-Na strani klijenta: ako podataka nema u lokalnom kesu salje upit lokalno podesenom DNS serveru

--udp/tcp port53

-Na strani servera: ako podataka nema u lokalnom kesu ili bazi zona salje se upit drugim DNS serverima autoritativnih za pripadajuci domen

-Dve vrste upita:

--Rekurzivni upit: DNS server u potpunosti vraca konacan odgovor ili gresku, upiti klijenata prema serveru

--Iterativni upit: DNS server vraca delimicni najbolji moguci odgovor, referise na druge servere u hijerarhiji koji mogu da dalje rese upit

@Definisanje zona

-Zona: txt ASCII fajl

-Resource record (RR)

--osnovna jedinica podataka - pojedinacan zapis u zoni

--- Name Time_to_live Class Type Value

----Name: domen/host adresa

----Time_to_live: vreme validnosti podataka u kesu u sekundama

----Class - Za internet uvek oznaka IN

----Type - tip RR podataka: SOA,NS,MX,A,AAAA,PTR...

----Value - vrednost koja se pridruzuje RR (adresa, naziv..)

-RR:SOA,NS,MX,A

@SOA zapis

-Start of Authority - def se na pocetku svake zone i sadrzi:

--naziv primarnog DNS servera - informativan podatak

--Email adresa DNS admina (tacka umesti @) - informativan podatak

--Serial - serijski broj zona fajla: yyyymmddnn, inkrementira se prilikom svake promene

--Refresh - posle koliko sekundi sekundarni DNS proverava primarni da li ima promena,da li je Serial povecan

--Retry - ako je neuspela prethodna provera, posle koliko sekundi se ponavlja

--Expire - koliko dugo u sekundama DNS cuva zone ucirane od primarnog DNS

--Minimum TTL - koliko dugo se rekordi iz zone cuvaju u lokalnom kesu drugih DNS servera

@NS,MX,A

-NS- polje definise autoritativne DNS servere za tekucu zonu ili poddomene

-MX- polje definise email server za tekucu zonu ili poddomene

-A - polje definise adresu za navedeno ime ili predefinisani server za tekucu zonu

@Glue record

-IP adresa DNS servera poddomena, definisana u zoni domen

--ako se za poddomen navodi DNS server preko imena, obavezno mora da bude definisana i IP adresa tog DNS servera (mora se nekako raresiti)

@CNAME - cannonical name

-uvodjenje alternativnih naziva za vec definisane nazive (alias)

-prednosti: definisanje samo 1 A zapisa i vise CNAME zapisa

-mane: upit za alias se razresava u 2 koraka sada

@PRT - inverzni DNS

-mapiranje IP adresa u nazive

-kreiran domen in-addr.arpa u kom su sve IP adrese u inverznom dotted-decimal formatu

--koriste se PTR tipovi resource record-a

@Zakup i odrzavanje domena, dns alati

-DNS provajderi - online kupovina i odrzavanje domena

-DNS alati- nslookup,dig

##
####################################

Ostali protokoli i tehnike na ruterima

_____

@Dodeljivanje IP adresa

-Staticko dodeljivanje IP adresa

--manuelno dodeljivanje fiksnih IP adresa

--ne mogu se podrzati ad-hoc korisnici - WiFi,VPN,..

-Dinamisko dodeljivanje IP adresa

--konfiguracija na jednom mestu - serveru (odredjeni opseg adresa)

--Protokoli automatske dodele adresaL RARP,BOOTP,DHCP

@RARP - reverse ARP

-nalazi se IP adresa na osnovu MAC adrese

-inicijalno bio namenjen za specificne uredjaje i radne stanice bez diska

-RARP server

--manuelno se definise mapiranje MAC adresa u odredjene IP adrese

--radne stanice po ukljucivanju pronalaze RARP server (broadcastom)

--RARP server na osnovu MAC adrese uredjaja pronalazi uparenu IP adresu

-Osnovni nedostaci: ne dodeljuje se maksa i default gateway, komunikacija samo na nivou L2 segmenata, ne i sa drugim mrezama

@ARP/RARP format zaglavlja

-protokol L3 nivoa

--poruke se enkapsuliraju u Ethernet okvire

-format zaglavlja kao kod ARP-a

--polje Operation - odredjuje ARP ili RARP funkcije

@RARP - princip rada

1.korak - RARP request - uredjaj po ukljucivanju generise RARP request paket i salje na broadcast

2.korak - RARP reply - RARP generise odgovor - paket sadrzi dodeljenu IP adresu i salje se na unikast MAC adresu uredjaja koji je inicirao zahtev. On je prihvata i pocinje da je koristi.

@BOOTP - Bootstrap Protocol

-dodeljuje IP adrese na osnovu MAC adrese (namena kao i RARP)

-BOOTP server - manuelno se definise mapiranje MAC adresa u odredjene IP adrese

-Dodatno, on moze da posalje i Default gateway, maskum dns server...

-Protokol aplikativnog nivoa, koristi UDP (67 ka serveru, 68 ka klijentima)

-osnovni nedostatak: staticko dodeljivanje IP adrsa na osnovu MAC adresa --> potrebno unapred poznavati MAC adrese korisnika, manuelna konfiguracija za sve korisnike,ne moze WiFi...

@BOOTP - princip rada

-Korak 1 - BOOT-request poruka - sadrzi MAC adresu posaljioca, enkapsulira se u UDP poruku, odredisni port 67. UDP se enkapsulira u IP poruku, brodkast odredisna IP adresa. IP se enkapsulira u Ethernet okvir, brodkast odredisna MAC adresa

--Svi uredjaji primaju request paket na L2 i L3 nivou, prosledjuju ga na L4 nivo, BOOTP server preuzima poruku na UDP portu 67, ostali je odbacuju

-Korak 2- BOOT-reply poruka - BOOTP server za MAC adresu se iz tabele mapiranja nalazi IP adresu, kreira se reply poruka koja sadrzi dodeljenu IP adresu i masku uredjaja, default gateway, IP adresu dns servera, ip adresu tftp servera... Ova poruka enkapsulira se u UDP poruku odredisni port 68. UDP se enkapsulira u IP poruku, brodkast odredisna IP adresa. IP se enkapsulira u Ethernet okvir, unikast MAC adresa uredjaja! Svi primaju poruke na l2 i l3 nivou, samo odredisni uredjaj slusa na UDP portu 68.

@DHCP- Dynamic Host Configuration Protocol

-Slicnosti sa BOOTP:

--UDP portovi 67 i 68

--slican format poruka

--slanje dodatnih parametara

-Razlike u odnosu na BOOTP:

--dinamicko dodeljivanje IP adresa iz predefinisanog opsega

--ogranicen period vazenja

--osim adrese i maske moze se dodeliti jos parametara

--moze da postoje vise DHCP servera u 1 mrezi

@DHCP - princip rada

-1.korak - DHCP-DISCOVER poruka - uredjaj po ukljucivanju generise tu poruku. Ona sadrzi MAC adresu posiljaoca. Enkapsulira se u UDP poruku, odredisni port 67. UDP se enkapsulira u IP poruku, brodkast odredisna IP adresa. IP se enkapsulira u Ethrenet okvir, brodkast MAC adresa. Svi uredjaji primaju DHCP-DISCOVER paket na l2 i l3 nivou, prosledjuju ga na l4 nivo. Svi dhcp serveri preuzimaju poruku na UDP portu 67, ostali je odbacuju.

-2.korak - DHCP-OFFER poruka - DHCP server sprovodi sledece: dodeljuju slobodnu IP adresu iz opsega rezervisanih IP adresa. Generise DHCP-OFFER poruku: izabrana IP adresa,maska,defult gateway,dns, drugi opcioni parametri. Poruka se enkapsulira u UDP paket, IP brodkast i MAC okvir (brodkast ili unikast zavisi od implementacije). DHCP server moze da proveri da li je neka adresa zauzeta slanjem ICMP ping paketa. Drugi DHCP server (ako postoji) na isti nacin salje svoju dhcp-offer poruku, nezavisno bira slobodnu op adresu koja moze da bude razlicita. Samo pocetni uredjaj prima okvire i preuzima dhcp-offer poruku.

-3.korak - DHCP-REQUEST

-pocetni uredjaj prihvata IP adresu iz PRVE dhcp-offer poruke

-dhcp-request poruka: sadrzi IP adresu koja se zahteva za koriscenje, kao i ostale parametre. Enkapsulira se u UDP poruku, odredisni port 67. UDP se enkapsulira u IP poruku, brodkast odredisna IP adresa. IP se enkapsulira u Ethernet okvir, brodkast odredisna MAC adresa. Oba DHCP servera prihvataju DHCP-REQUEST poruku --> sinhronizacija iskoriscenjih IP adresa

-4.korak - DHCP-ACK - svi DHCP serveri prihvataju DHCP-request poruku. Samo DHCP server koji je ponudio zahtevanu IP adresu generise potvrdu - dhcp ACK poruku. Poruka se salje na brodkast IP i unikast ili brodkast MAC u zavisnosti od implementacije. IP adresa se iznacava kao iskoriscena na odredjeno vreme. Pocetni uredjaj prihvata DHCP-ACK poruku i pocinja da koristi dobijene parametre.

-Implementacija: DHCP server - NA SVAKOJ LAN mrezi (brodkast domen)ili DHCP na ruteru - za sve pripadajuce LAN mreze

@NAT - NETWORK ADDRESS TRANSLATION

-privatne IP adrese: 10.0.0.0/8, 172.16.0.0/12,192.168.0.0/16

--stede potrosnju javnih IP adresa, ne smeju da se oglase na Internetu

-Sprovodimo translaciju adresa - pretvaranje IP adresa iz jednog skupa adresa u drugi. NAT se sprovodi na granicnom ruteru (jedinstvena tacka povezivanja sa ostatkom mreze)

@NAT terminologija

-Inside Local Address - adresa dodjeljena hostu na unutrasnjoj mrezi cije se adrese transliraju

-Inside Global Address - legitimna IP adresa dodeljena pd strane provajdera. Adresa u koju se pretvara INside Local adresa

-Outside Global Address - IP adresa uredjaja na spoljasnjoj mrezi

@Staticki NAT

-fiksno mapiranje "jedan jedan" - jedna lokalna uvek u istu globalnu

-NAT tabela - unapred definisana pravila mapiranja - parovi lokalne i globalne adrese

--uobicajeno ponasanje klijenti unutra, serveri spolja

-Proces na ruteru:

--Iz unutrasnje ka spoljasnjoj: izvorisne lokalne adrese iz zaglavlja IP paketa se pretvaraju u globalne

--Iz spoljne ka unutrasnjoj: odredisne globalne adrese iz zaglavlja IP paketa se pretvaraju u lokalne

-Prednost: inicijalizacija komunikacije iz spoljne mreze ka unutrasnjoj

-Nedostatak: ne postize se puna usteda adresa

@Dinamicki NAT

-definise se pool tj skup IP adresa

-Pri komunikaciji iz unutrasnje mreze uzima se slobodna adresa

-NAT tabela se dinamicki popunjava parom lokalne i globalne adrese

-U jednom trenutku jednu globalnu adresu moze koristiti samo jedna lokalna --> maksimalni broj konekcija je broj adresa u NAT pulu

-Tokom vremena - po zavrsetku komunikacije brise se korisceno mapiranje - oslobadja se adresa. Dakle vise unutrasnjih adresa se moze mapirati u manji broj globalnih adresa,

-Konekcije se iniciraju samo iz unutrasnje mreze ka spoljasnjoj

-Oslobadjanje globalne adrese kada se komunikacija zavrsi

--TCP-moze se prepoznati kada se sesije regularno zatvaraju, sta ako se sesija nasilno prekine?

--UDP-ne zna se koliko ce da draje i da li ima jos paketa iako se ne koristi

--ICMP - kratkotrajne sesije

->Uvodi se tajmer za svaki red u NAT tabeli. Red se brise nakon isteka tajmera

@Overload NAT

-Kako vise lokalnih adresa ISTOVREMENA da koristi manji broj globalnih adresa? Potrebne dodatne informacije za obezbedjivanje jednoznacnosti -> KOristi se TCP i UDP port - PAT (PORT ADDRESS TRANSLATION) - soket ga cini jedinstvenim

-Klijent mora da bude u unutrasnjoj mrezi -> klijentski port se slucajno bira na strani klijenta (proizvoljan), pa moze i da se promeni prilikom nat-a

@PAT sa jednom globalnom adresom

-moze da se koristi i samo jedna globalna adresa, broj porta pravi distinkciju

@Port forwarding

-Kako za server u unutrasnjoj mrezi omoguciti pristup iz spoljasnje? - problem

-Port-forwarding - staticko mapiranje za odredjene adrese i portove

--spolji zahtev na globalnu IP adresu i serverski port ce se mapirati u lokalnu IP adresu servera, a serverski port ce ostati nepromenjen. OMogucava da se serveru pridje iz spoljasnje mreze.

--nedostatak: moze samo jedna lokalna IP adresa da bude uparena sa serverskim portom, odnosno samo jedan server za svaki servis (port)

@Direktna komunikacija preko NAT-a:

-Dinamicki NAT/PAT: inicijalizacija komunikacije - iz unutrasnje mreze

-Kako se sprovodi direknta komunikacija 2 uredjaja u razlicitim unutrasnjim mrezama (iza NAT-a):npr real time aplikacije poput skype,viber,...

-Resenje: uredjaji se najpre registruju na javno dostupnom serveru. Server uredjajima prosledi NAT-ovane adrese i portove za pristup i uredjaji nastavljaju direktno da komuniciraju.

->Ove sve vazi za UDP

--Problem: kako direktna TCP veza? Kako NAT dozvoljava 3.uredjaju da koristi prethodno mapiranu adresu?

@NAT i UDP

-TCP-komunicira se samo sa uredjajem sa kojim je prethodno uspostavljena konekcija, dok kod UDP nema otvaranja konekcije pa na adresu i port moze svako da pristupi, cak i na klijentski ako ga poznaje.

-Kome se dozvoljava da koristi globalnu adresu i port za otvorene NAT konekcije?

--4 slucaja: - kome dozvoljava da udje

---symmetric - najrestriktivnije - samo taj uredjaj i taj port

---full-cone - najmanje restriktivno -bilo koji uredjaj i bilo koji port

---restricte-cone - samo taj uredjaj i svi portovi

---port-restricted-cone - svi portovi i na drugim uredjajima

@NAT i ICMP i ostale aplikacije

-kako radi NAT za ICMP pakete koji ne koriste UDP/TCP poruke?

->Dve vrste poruka:

-poruke upita: poseduju identifikaciono polje koje se koristi za NAT mapiranje

-poruke o gresci: ne poseduju identifikaciono polje, originalni IP paket se prenosi u telu ICMP poruke - > potrebno je promeniti lokalne adrese i portove i u originalnom paketu

-Pojedine aplikacije prenose informacije o IP adresama u svojim podacima

--Application Level Gateway (ALG)

---NAT uredjaj mora da gleda i menja i aplikastivne podatkem kako bi NAT bio transparentan

@Koriscenje NAT-a

-Prednosti: privatne korporacijske mreze koriste privatne IP adrese -> slobofa u dodeli i koriscenju; ne mora da se vrsi promena adresa u privatnoj mrezi prilikom promene provajdera; povecana je sigurnost (dinamicki NAT) jer je priv deo mreze izolovan; manja je potrosnja javnih IP adresa

-Mane: slozenija konfiguracija i administracija, otezano pracenje dogadjaja...

@ACL - Access Control Lists - kontrola prosledjivanja paketa

-dozvola ili zabrana prolaska paketa kroz interfejse rutera

-inspekcija zaglavlja na L3 i L4 nivou

--uslov: poredjenje IP adresa, TCP/UDP portova,ICMP poruka

-Akcija

--dozvola (Permit) - propustanje paketa

--zabrana (Deny) - odbacivanje paketa (unistavanje) - slanje na Null interfejs

-Filtriranje paketa - Packet FIltering

-Za svaki paket: prolazak kroz uredjenu listu uslova i pravila

-Nailazak na prvi ispunjen uslov -> izvrsava se pridruzeno pravilo i zavrsava se prolaz kroz listu

-Kraj liste - ni jedno pravilo nije ispunjeno -> paket se odbacuje

-Primena na interfejs: na ulasku u interfejs (IN); na izlasku iz interfejsa (OUT)

##
##################################

IPv6

_____

@Uvod

-IPv4 - problemi: nedostatak adresnog prostora, velike tabele rutiranja

--nove potrebe: bezbednost podataka na IP nivou, ostvarivanje kvaliteta servisa QoS

-Osnovne karkateristike IPv6: veci adresni prostor, efikasnije rutiranje jer ima manji broj eksternih ruta zbog HIJERARHIJSKE strukture mreznih adresa na Internetu i jednostavnije je zaglavlje

-Podrska za automatsku konfiguraciju racunara

-Podrska za bezbednost podataka sa IPSec implementacijom

-Poboljsana podrska za mobilne uredjaje

-Ugradjena podrska za alokaciju resursa i kvalitet servisa

-POvecan broj multicast adresa

@Format zaglavlja

-Izbacena polja:

--INternet Header Length - kod ipv6 zaglavlje je fiksne velicine

--Header Checksum - provera integriteta paketa se sprovodi na L2 nivou

--Options - nedovoljno se koriste u ipv4 dok u ipv6 imamo fleksibilno ugnjezdavanje opcija u dodatnim zaglavljima

--Polja za fragmentaciju

@Fragmentacija

-sprovodi se na izvoristu, ne u ruterima

-MTU - Maximum Transmission Unit - ipv6 garantuje MTU od min 1280 B. U slucaju da ruter ne moze da prosledi paket jer je veci od MTU on se unistava i ruter generise poruku 'Packet Too Big'

-Path MTU Discovery - pronalazi najmanji MTU na celom putu do odredista

--salje pakete odredjene velicine i prati da li je dobio "Packet Too Big'

-Problem: rutiranje je dinamicko i putanja se moze promeniti tokom komunikacije

@IPv6 format zaglavlja

-Traffic Class (8b)

--isto kao ToS kod IPv4

--izvoriste generise pakete koji pripadaju razlicitim klasama

-Flow Label (20b)

--tok=komunikacija izmedju aplikacija izvorista i odredista

--jedinstveno oznacava svaki tok

--samo se prvi paket rutira, FLow Label uparen sa izlaznim portom se kesira

--ubrzan proces rutiranja

-Payload Length(16b) - duzina podataka u bajtovima

-Hop Limit (8b) - isto kao TTL

-Next Header (8b) - umesto polja Protocol, identifikuje sledece zaglavlje tj zaglavlje viseg nivoa, zaglavlje sa ipv6 opcijama

@Ruting opcija

-Utice na put paketa - izvoriste definise sekvencu rutera tj checkpoints

--zaglavlje ruting opcije sadrzi sekvencu adresa medjutacaka i brojac

-Princip rada: izvoriste definise sekvencu adresa medjutacaka,poslednja adresa je odrediste. Odredisna adresa regularnog IPv6 zaglavlja je adresa prve medjutacke. Ruter kada prepozna sebe kao odrediste, a postoji ruting zaglavlje radi N=segment left, adresa odredista se menja sa adresom na N-toj poziciji od kraja sekvence

@ipv6 adrese

-duzine 16B tj 128b --> 4 puta vece od ipv4

-pise se u heksadekadnom obliku - jedna heksadekadna cifra od 4b, razdvojeni dvotackom

-skkraceni zapis - izbaciti vodece nule u grupama od 4 cifre 00x => :x

---izbaciti SAMO jedan niz grupa sa nulama :0:0:0: => ::

@vrste ipv6 adresa

-unikast adresa - jedinstvena, identifikuje interfejs

-multikast - adresa koja identifikuje vise interfejsa razlicitih uredjaja prema nekoj zajednickoj nameni

-anykast-adresa koja identifikuje vise interfejsa razlicitih uredjaja ali paket poslat na anycast bice prosledjen SAMO JEDNOM interfejsu - bilo komw nw svima

@Unicast adrese

-Global Unicast Address 2000::/3

-Unique Local Address fc00::/7

-Link-Local Address fe80::/10

@Global Unicast adrese - javne adrese

-dostupne na Internetu

-pocinju binarnom vrednosti 001

-terminologoja: Prefix - mrezni deo, Interface ID - adresa interfejsa (kao host za ipv4)

-logicki se deli na 3 dela:

--globalni prefiks (tipicno prvih 48b koji se dodeljuje provajderima..)

--adresa podmreze (tipicno 16b)

--adresa hosta odnosno interfejsa - tipicno poslednja 64b

-Mrezni deo, tj maska moze da udje i u interfejs ID ali to nije dobra praksa

@Global ROuting Prefix, SUbnet ID

-Agregacija prefiksa - hijerarhijska raspodela - kontinetni (RIR), globalni provajderi... -> efikasnije rutiranje

-Provider-Aggregatable (PA) - pripada opsegu adresa provajder-agregacija

prednost: ne naplacuje se, provajderi dodeljuju svojim korisnicima, ali promena provajdera zahteva promenu prefiksa

-Provider-Independent (PI) - dodeljuje se od strane RIR-a, kao i mreze provajdera. Prednost - nezavisne od provajdera ali se obicno naplacuju

@Interface ID

-polje adrese koje oznacavaju uredjaje u podmrezi

-postavljanje:

--staticki - manuelno - proizvoljna vrednost, obicno mali brojevi, dozvoljene su i sve jedinice i nule

--dinamicki - automatski - random ili pravilo EUI-64

@EUI-64 - Extended Unique Identifier

-pravilo generisanja INterface ID na osnovu MAC adrese

--6B MAC adrese se prosiruje na 8B koji cine intergace ID

--deli se MAC adresa na 2 grupe od po 3B

--u sredini se umecu 2 bajta ff i fe

--sedmi bit prvog bajta u/l (universal/local) postavlja se na 1: 0- Universal gde je MAC burned-in ili 1- Local: MAC je logicki konfigurisana na proizvoljan nacin i ima lokalno znacenje

@Unique Local Address

-privatne adrese

-opseg fc00::/7

--sedmi bit: 0-trenutno se ne koristi, 1- trnutno jedino dozvoljeno

-ne smeju da se oglasavaju na internetu

-Global ID - pseudo-slucajna vrednost

@Link Local adrese

-za koriscenje samo unutar lokalne IP mreze (L2 segment)

-opseg: fe80::/10

-nov koncept

-ruteri ne prosledjuju pakete sa ovim adresama

-dodela interfejs ID - automatski EUI-64 ili manuelno ili random

@Specijalne Unicast adrese

-Loopback Address ::1/128 - logicka adresa za lokalne koriscenje na jednom uredjaju tj adresa tog uredjaja kao 127.0.0.1 i ne izlazi van uredjaja tj ne rutira se

-Unspecified Address - ::/128 - npostojeca, sadrzaj adresnog polja kada adresa nije poznata - samo kao izvorisna adresa koja se ne rutira

-Embedded ipv4 address - ::/80 sa ::ffff:192.168.10.10

@ipv6 multicast adrese

-opseg ff00::/8

-flags: 4 bita specijalnih flegova

--T fleg: 0-Well-Known - predefinisane permanentne adrese dodeljene od IANA; 1-Transient - dodeljene po potrebi od strane razlicitih multikast aplikacija

--Scope - 4 bita koja definisu opseg koriscenja: 2- samo na lokalnom L2 segmentu; 8- na nivou organizacije (Subnet ID) i 14 (E) - globalni opseg

@Solictited-Node Multicast adrese

-Odnose se na pojedinacne uredjaje, automatski generisane iz unicast adresa

-koriste se za internu komunikaciju - Neighbor DIscovery Protocol

--Address Resolution (kao ARP)

--Duplicate Address Detection (DAD)

@Anycast adrese

-identifikuje vise interfejsa koji pripadaju i razlicitim uredjajima, bilo gde na mrezi

--VISE UREDJAJA SA ISTOM ADRESOM NA MREZI

--paketi stizu SAMO DO JEDNOG najblizeg uredjaja (odredjeno protokolom rutiranja)

-Nije poseban opseg adresa, vec koncept rutiranja unicast adresa

@Konfigurisanje IPv6 adresa

-Staticko- konfigurisanje cele adrese - 128b - neprakticno -> konfiguracija samo mreznog dela adrese tj 64b -> interface id se automatski postavlja po pravilu EUI-64

-Dinamicko

--Stateful DHCPv6 - po analogiji za ipv4 - samo jos i pamti kom uredjaju je dodelio koju adresu

--Stateless Address Autoconfiguration (SLAAC) - automatsko uspostavljanje, ugradjena fja u ipv6, uredjaji automatski saznaju mrezni deo od 64b, default gateway i dns server (opciono),interface id po eui-64

@ICMPv6 - internet control message protocol

-Error Messages

--Destination Unreachable

---Network Unreachable

---Address Unreachable

---Port Unreachable

---Reject route to destination

--Packet Too Big

--Time Exceeded

--Parameter Problem

-Informational Messages

--Ping

--Multicast Listener Discovery

--Neighbor Discovery Protocol

@Neighbor Discovery Protocol

-NDP zamenjuje ARP,ICMP Router Discovery i ICMP Redirect

-Funkcije:

--Router discovery -otkirvanje svih povezanih rutera

--Prefix discovery -otkrivanje mrezne adrese

--Address Resolution - kao ARP

--Duplicate Address Detection -da li je adresa iskoriscena

--Redirect -kao ICMP redirect

--Neighbor Unreachability Detection

-Poruke: Router Solicitation (RS) i Router Advertisment (RA); Neighbor Solicitation (NS) i Neighbor Advertisment (NA)

@Autokonfiguracija ipv6 uredjaja

-Stateless Address Autoconfiguration (SLAAC)

-Koristi se NDP u 2 koraka

-1. korak - uredjaj salje upit ruteru preko Router Solicitation poruke

--uredjaj salje upit svim ruterima na lokalnoj mrezi

--izvorista IP adresa: Link-Local adresa uredjaja

--Odredisna IP adresa: multikast adresa FF02::2 (Svi IPv6 ruteri)

-2.korak - ruter odgovara slanjem Router Advertisment (RA) poruke

--interfejs rutera je konfigurisan sa unikast IPv6 adresom

--RA poruka, kao odgovor na RS:

---izvorisna IP adresa: Link-Local adresa rutera

---odredisna IP adresa: Link-Local adresa uredjaja

---sadrzaj: mrezna adresa (prefiks), opciono i DNS

---Default Gateway- uzima se izvorisna IP adresa (Link-Local adresa rutera)

--Uredjaj sam odredjuje INterfejs ID koristeci EUI-64 ili random pravilo

-Nezavisno od RS poruke, ruteri periodicno samostalno oglasavaju RA poruke

--Izvorisna adresa IP adresa: Link-Local adresa rutera

--Odredisna IP adresa: FF02::1 (All IPv6 Devices)

--period oglasavanja: Cisco ruteri 200 sekundi

@Dodela DNS servera

-Slanje adrese DNS server - opciono polje u RA poruci

-Alternative:

--Stateless DHCPv6- dodeljuje samo DNS server, ali ne i IP adrese; ne pamti sta je poslato pojedinacnim uredjajima

--Statefull DHCPv6 -dodeljuje se IPv6 adresa,maska,default gateway,DNS server i ostali parametri; pamti se koje su adrese dodeljene

-DNS server - AAAA zapis - IPv6 adrese

@Address Resolution

-Kao ARP protokol - za poznatu IPv6 adresu trazi se MAC adresa

-1.korak-uredjaj salje upit preko Neighbor Solicitation (NS) poruke

--salje bilo koji uredjaj kome je potrebna MAC adresa

--izvorisna adresa: unikast adresa uredjaja koji zahteva MAC adresu

--odredisna adresa: Solicited-Node Multicast adresa uredjaja za poznatu IP adresu

--odredisna MAC adresa: multikast

-2.korak - prozvani uredjaj salje Neighbor Advertisment (NA) poruku

--odgovara uredjaj sa navedenom Solicited-Node Multicast adresom iz NS poruke

--izvorisna adresa: unikast adresa uredjaja koji salje NA

--odredisna adresa: unikast adresa uredjaja koji je poslao NS poruku

--sadrzaj: zahtevana MAC adresa uredjaja koji salje NA poruku

@Duplicate Address Detection (DAD)

-automatski se sprovodi za rucno postavaljene adrese,SLAAC,DHCPv6

-salje se upit za adresu koja se zeli koristiti (NS) i ceka da li ce neko da odgovori (NA)

@IVP6 protokoli rutiranja

-destination-based

-longest-prefix match

-Protokoli rutiranja: RIPng,OSPFv3,IS-IS,Multiprotocol BGP

@IPv4 i IPv6

-mehanizmi tranzicije sa 4 na 6:

--IPv4/IPv6 Dual Stack

--IPv6 Tunelovanje - enkapsulacija IPv6 paketa u IPv4 paket

--Mehanizmi translacije protokola - omogucava komunikaciju ipv4 i ipv6 uredjaja

@IPv4/IPv6 Dual Stack

-Dvostruki IP sloj- svi protokoli visihh slojeva mogu komunicirati preko ipv4 i ipv6; pojedine aplikacije ce prvo pokusati komunikaciju preko ipv6

@IPv6 Tunelovanje

-Mogucnost komunikacije IPv6 mreza koje su povezane preko IPv4 mreze

-Na prelasku iz IPv6 u IPv4 - IPv6 paketi se enkapsuliraju u IPv4 pakete a posle dekapsuliraju

@Mehanizmi translacije protokola

-direktna komunikacija izmedju IPv6 i IPv4 uredjaja

-NAT-PT

-SLikcno kao i IPv4 NAT - zamena adresa u zaglavlju na granicnom ruteru izmedju ipv6 i ipv4 domena