ETHERNET

4 osnovna IEEE standarda:

IEEE 802.3 - standard za Ethernet

IEEE 802.11 – standard za WiFi

IEEE 802.15 – Bluetooth

IEEE 802.16 - WIMAX

Ethernet radi na poslednja dva sloja OSI modela: Data Link i Fizicki sloj. Nije definisan na mreznom sloju, zato sto IP paket izgleda isto bez obzira preko koje se tehnologije prenosio na L2.

Ethernet je prvo objavljen 1980. godine, a standardizovan je 1985. godine. Postoje i razlike u zaglavlju ethernet frejma.

Layer 2 Addresses Layer 1 Limitations

Layer 1 Limitations	Layer 2 Functions
Cannot communicate with upper layers	Connects to upper layers via Logical Link Control (LLC)
Cannot identify devices	Uses addressing schemes to identify devices
Only recognizes streams of bits	Uses frames to organize bits into groups
Cannot determine the source of a transmission when multiple devices are transmitting	Uses Media Access Control (MAC) to identify transmission sources

Razlike izmedju L1 i L2 se najbolje vide u razlikama izmedju Switcha, uredja L2 slojam i HUB-a, uredjaja L1 sloja.

Data Link sloj se sastoji od dva podsloja:

1) LLC – zadatak da komunicira sa mreznim slojem, da poveze softver sa hardverom

2) MAC – zadatak da smjesti frame u odgovorajuci trenutak na slobodni medijum

Ethernet se razvio iz protokola koji se zvao Aloha net, koji je nastao 1970. Godine kao ideja da se povezu Havajska ostrva preko radio talasa.

Prvi nacin za povezivanje racuna u lokalne mreze je bila magistrala(bus).

Magistrala je bio kanal gdje su se svi racunari kacili preko koaksijalnih kablova. Glavni problem magistrale je bio to da ako jedan kabl otkaze ,citava mreza pada. Ovo je bila jako nestabilna topologija, tako da su uradjena dva bitna unapredjenja.

Prvo se sa magistrale preslo na zvijezdu, odnosno imali smo uredjaj HUB koji je omogucio da ako otkaze jedan racunar, ostali mogu da nastave da komuniciraju. Presli smo sa koaksijalnih na UTP kablove. UTP kablovi su mnogo fleksibilniji za koristenje od koaksijalnih kablova.

Prva dva koaksijalna kabla koja su se koristila u lokalnim mrezama su:

10BASE5 (Thicknet) & 10BASE2 (Thinnet)

Kod naziva kablova prvi broj predstavlja brzinu koja se moze ostvariti pomocu tog kabla. Ovi koaksijalni kablovi su imali 10Megabitnu brzinu.

I kod UTP kablova prvi broj oznacava brzinu koja se moze ostvariti pomocu tog kabla.

Da bi bila veca brzina koja se moze ostvariti kod UTP kablova, mora se kompleksnije uraditi kodovanje.

Poslednji broj u nazivu koaksijalnih kablova je duzina na kojoj ovi kablovi mogu da rade prije nego sto treba da se uradi regenerisanje signala. Ova 5 kod Thicknet kabla predstavlja 500 metara, tako da on moze da radi na 500 metara, prije nego sto je potrebno uraditi regeneraciju signala.

Kod UTP kablova je 100 metara ta duzina, a kod optickih moze da bude I vise kilometara.

Thinnet ova duzina je 200 metara.

Nakon koaksijalnih kablova preslo se na UTP kablove koji su jeftiniji I laksi za rad, I danas se koriste za Ethernet u lokalnim mrezama.

Danas su najcesci UTP kablovi, ali ako zelimo da ostvarimo 10Gigabitnu brzinu moraju se koristiti STP kablovi(Shielded Twisted Pair).

Prvi UTP kablovi koji su se pojavili su bili 10BASE-T , T oznacava da se radi o UTP kablu. Kod ovih kablova se koristilo Manchester kodovanje. Prednost Manchester kodovanja u odnosu na Non-Return to Zero je

to sto se ovdje koduje rast ili pad signala I samim tim smo oneogucivali da se pojavi dugacka prava linija, koja moze da narusi funkcionalnost.

Danas su najcesci 100BASE-TX kablovi, to su 100 Megabitni kablovi(Fast Ethernet) I danas se uglavnom koristi Switch umjesto HUBa.

Razlika izmedju Full-Duplexa I Half-Duplexa je sto full-duplex omogucava komunikaciju u oba smjera istovremeno, dok half-duplex omogucava komunikaciju u oba smjera, ali ne istovremeno.

Kodovanje kod 100BASE-TX kablova je 4B5B, gdje smo svaka 4 bita zamjenjivali sa drugih 5 bitova, I tako saljemo vise podataka, ali dobijamo sigurniju prenos.

Razlika izmedju sifrovanja i kodovanja je to sto kod kodovanja imamo kodnu knjigu, odnosno moramo da za svaku ulaznu sekvencu da definisemo sta je izlazna sekvenca. Ako nije predefinisan ulaz, nikako ne mozemo da odredimo izraz. Kod sifrovanja ulazna sekvenca moze biti bilo sta I dobice se izlazna sekvenca.

Gigabitni Ethernet mozemo ostvariti preko UTP kablova, ali moramo imati kategoriju 5E ovih kablova, ali da bismo ostvarili 10Gigabitni prenos moramo da imamo STP kablove, ne mozemo to da uradimo pomocu UTP kablova.

Types of Ethernet

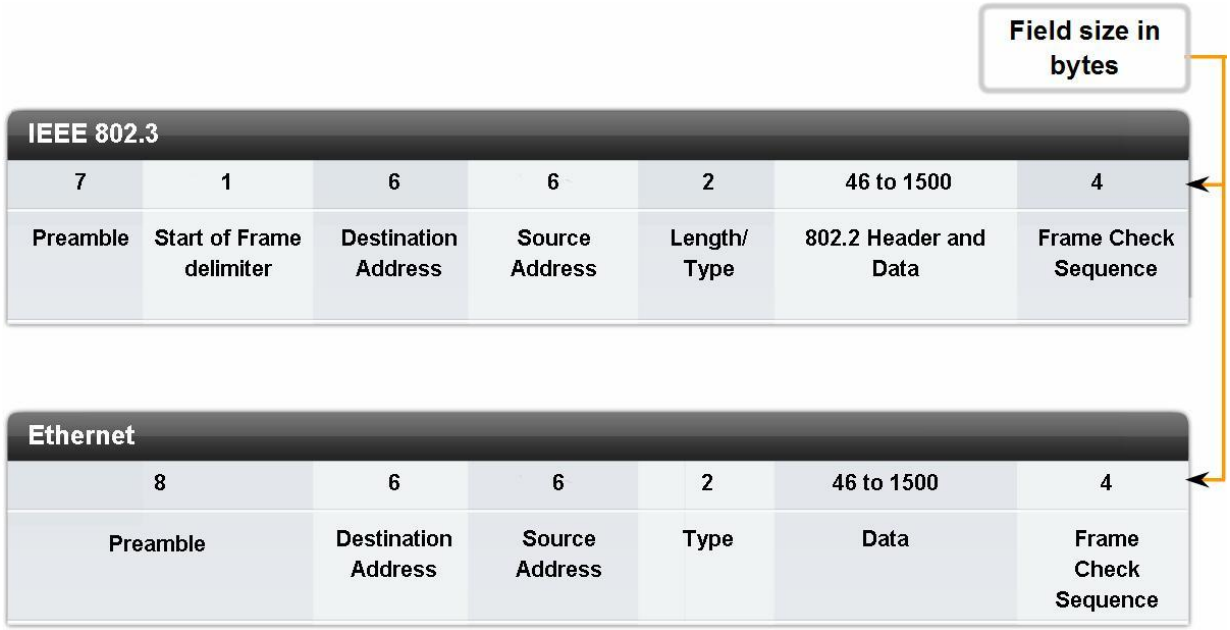
Ethernet Type	Bandwidth	Cable Type	Duplex	Maximum Distance
10Base-5	10 Mbps	Thicknet Coaxial	Half	500 m
10Base-2	10 Mbps	Thinnet Coaxial	Half	185 m
10Base-T	10 Mbps	Cat3/Cat5 UTP	Half	100 m
100Base-TX	100 Mbps	Cat5 UTP	Half	100 m
100Base-TX	200 Mbps	Cat5 UTP	Full	100 m
100Base-FX	100 Mbps	Multimode Fiber	Half	400 m
100Base-FX	200 Mbps	Multimode Fiber	Full	2 km
1000Base-T	1 Gbps	Cat5e UTP	Full	100 m
1000Base-TX	1 Gbps	Cat6 UTP	Full	100 m
1000Base-SX	1 Gbps	Multimode Fiber	Full	550 m
1000Base-LX	1 Gbps	Single-Mode Fiber	Full	2 km
10GBase-CX4	10 Gbps	Twin-axial	Full	100 m
10GBase-T	10 Gbps	Cat6a/Cat7 UTP	Full	100 m
10GBase-LX4	10 Gbps	Multimode Fiber	Full	300 m
10GBase-LX4	10 Gbps	Single-Mode Fiber	Full	10 km

Prvi broj uvijek oznacava brzinu prenosa koja se moze ostvariti, drugo je uglavnom BASE-prenos se vrsi u osnovnom opsegu, a poslednji znaci oznacavaju na koji se tip kabla odnosi.Ako je T kao poslednje slovo, to se radi o UTP kablu, a ako imamo F kao fiber ili L kao light, onda se uglavnom radi o optickim

kablovima. Ovdje pise jos da li je Full ili Half duplex kao I maksimalna distanca nakon koje je potrebno izvrsiti regeneraciju signala.

Razlika izmedju Single mode I Multimode kablova je sto kod Single moda koristimo laser kao izvor svijetlosti koji je pouzdaniji nacin, dok kod multimode koristimo diodu.

Polaj u Ethernet frejmu:



Prva slika je originalni Ethernet iz 1980. dodine dok je druga slika izgled frejma nakon revizije 1985. godine.

Prvih 8 bajtova je Preambula I koristi se za sinhronizaciju ucesnika u komunikaciji, I ta preambular se nikad ne smatra dijelom frejma.

Kad govorimo kolika je velicina frejma, nikad ne uzimamo u obzir preambulu. Preambula ne ulazi u frejm, ne ulazi u velicinu frejma, niti bajtovi preambule ulaze kad racunamo CRC vrijednost koja se smjesta u FCS polje. Ovih 8 bajtova se iskljucivo koristi za sinhronizaciju.

U reviziju su Preambulu podijeli na dva dijela, 7 bajtova Preambule I jedan bajt koji predstavlja Start of Frame delimiter, ali to je prakticno isto I nebitno.

Poslije ovog dolaze Mac adrese. Mac adresa je velicine 6 bajtova, tako da sun am potrebni 12 bajtova za Source I Destination Mac adrese. Ovdje je karakteristican redoslijed Source I Destination adresa. Kod svih prethodnih zaglavlja prvo se upisivao podatak ko salje, a onda ko prima. Ovdje je obrnuto, prvo se nalazi Destination, pa tek onda Source adresa. Ovo je tako jer postoje dvije vrste switcheva. Jedna od tih vrsta su brzi switchevi koji mogu da dodnesu odluku sta ce sa Frejmom cim prime prvih 6 bajtova tog frejma. Kad bi destination adresa bila na drugom mjestu, switch bi morao da primi prvih 12 bajtova da bi donijeli odresu sta ce sa frejmom. Ovo je uradjeno iskljucivo zbog brzine, da bi brzi switchevi mogli brze da donesu odluku sta da rade sa primljenim frejmom.

Nakon ovog ide polje koje se zove Length/Type. U revizij ije omoguceno da ovo polje govori dvije razlicite informacije, odnosno istovremeno ne moze govoriti dvije informacije, nego ili jednu ili drugu, biramo sta se koristi. Druga strana zna da li saopstavamo jednu ili drugu stvar, Length je velicina cijelog frejma, a Type oznacava tip protokola koji je enkapsuliran u frejm.

Poslije toga ide Paket koji silazi sa mreznog sloja. Velciina paketa je od 46 do 1500 bajtova. Ako se pojavi paket koji je veci od 1500 bajtova onda moram da vrsimo fragmentaciju jer u jedan Ethernet frejm ne mozemo da smjestimo paket koji je veci od 1500 bajtova. Minimalna velicina paketa koji mozemo smjestiti u ovo polje je 46 bajtova, a maksimalna 1500.

Kontrolne informacije: 12 bajtova za adrese, 2 bajta za polje Type/Length I 4 bajta za FCS = 18 bajtova

Frejm sam moze da bude od 64 do 1518 bajtova. To su granice Ethernet frejma.

Ako se pojavi paket veci od 1500 bajtova, zadatak mreznog sloja je da ga podijeli na manje fragmente.

Ako se pojave podaci koji su manji od 46 bajtova,npr. Telnet protokol salje podatke bajt po bajt, paket koji se tako formira je velicine 41 bajt, a to je premalo da bi se smjestilo u frejm. Onda se desava padding – dodavanje besmislenih niza jedinica I nula koji je samo zadatak da dopuni paket, odnosno frejm do odgovarajuce velicine.

Length/Type ima 2 bajta, maksimalna velicina broja koji se moze smjestiti je 65535. Medjutim, maksimalni frejm je 1518 bajtova. Onda je dogovoreno da sve vrijednosti koji su izmedju 0-1518 onda je to Length, a ako se koriste veci brojevi, onda je to Type, na taj nacn je realizovano da jedno polje moze govoriti dvije informacije.

FCS ima svaki frejm, bez obzira kakav protokol se koristi, I ovdje se smjesta vrijednost CRC koja se koristi kao verifikacija da li je doslo do promjene prilikom slanja frejma.

Racunar kreira CRC, dolazi do rutera, ruter provjera da li je to taj frejm, ako jeste, uzima paket I kada bude ponovo enkapsulirao, racunace novu CRC vrijednost.

Frejm moze biti I veci od 1518, odnosno moze se desiti da na frejm dodamo jos 4 bajta kontrolnih informacija. To se desava ako koristimo VLANove, I tu dodajemo informaciju o kojem VLANu se radi.

600h, odnosno 1536 je granica izmedju Lengtha i Type. Manje je Length, vise je Type.

MAC adresa ima 48 bita. Te bite mozemo podijeliti u dva dijela. Ta dva dijela nemaju nikakve veze sa onim dijeljenjem kod IP adresa. MAC adresa se uglavnom nalazi na mreznoj kartici koja se nalazi u racunaru. Tu mreznu karticu je neko proizveo (Huawei, CISCO, IBM..), I prilikom proizvodnje te kartice posebnom procedurom je ugorena mac adresa u samu karticu. Ta Mac adresa je stalna, ne moze da se mijenja. To znaci da ako ovaj racunar prebacimo u neku drugu mrezu, Mac adresa ostaje ista. Mac adresa ima iskljucivo lokalno znacenje, ona vrijedi samo u mrezu u kojoj se nalazi racunara. Mac adrese racunara u nekoj drugoj mrezi meni nista ne znace.

IP adresiranje je hijerarhijsko, uvijek se mora da postuje IP adresa na osnovu toga gdje se nalazi racunar. MAC adresiranje flat (ravno), jer je ona uvijek ista bez obzira gdje se racunar nalazi.

To nam omogucava da prilikom slanja podataka u neku drugu mrezu kombinacije IP adresa ne mjenjamo, a MAC adrese mjenjamo od mreze do mreze.

MAC adresu pisemo pomocu heksadecimalnih brojeva.

Treba nam 12 razlicitih heksadecimalnih cifara za predstavljanje MAC adrese.

12-34-56-AB-CD-EF

1234.56AB.CDEF

1234:56AB:CDEF

Kako god je zapisali, MAC adresa ima dva dijela I uvijek je granica na pola.

Prva 24 bita se nazivaju OUI – Organizational Unique Identifier.

Ako imamo kompaniju koja zeli da proizvodi mrezne kartice, onda ce se ona obratiti nekoj krovnoj organizaciji I trazicu od nje da mi da OUI, OUI je jedinsveno za svaku organizaciju koja proizvodi mreznu karticu. Na osnovu prvih 24 bita se uvijek zna ko je proizveo mrezne kartice.

Ipconfig /all – komanda za prikaz MAC adrese.

Proizvodjac mreznih kartica je duzan da ostala 24 bita napravi jedinstveno.

Jedino je problem ako se dvije MAC adrese pojave u istoj lokalnoj mrezi, ako su dva racunara sa istim MAC adresama u razlicitm mrezama, onda nemamo problem, ali je svakako duznost onih kojji proizvode te mrezne kartice da proizvedu kartice sa jedinstvenim MAC adresama.

MAC adresu koristi I WiFi I Ethernet.

Tipovi slanja kod Etherneta:

1) unicast 1-1 : unicast uvijek saljemo u vasoj mrezei, cak I saljemo van mreze, na MAC nivou cete slati gatewayu u nasoj mrezi, Source MAC je nasa MAC adresa a destination MAC je Mac adresa uredjaja kojem saljemo, bilo to drugi racunar ili gateway

2)broadcast 1-svima FF-FF-FF-FF-FF-FF MAC broadcast mozemo uputiti samo u nasu mrezu, a kod IP je moguce izracunati broadcast adresu udaljene mreze I poslati broadcast poruku u tu mrezu

Dva glavna protokola koji koriste broadcast poruke su DHCP I ARP

3)multicast 1-prema nekoj grupi

Multicast MAC adrese uvijek pocinju sa prefiksom 01-00-5E. Ovaj prefix ne moze da dobije nijedna firma koja proizvodi mrezne kartice. Multicast MAC adresa se direktno dobija od multicast IP adese. Multicast IP adrese pocinju od 224-239 u prvom oktetu.

Preostala 24 bita MAC adrese dobijamo na sledeci nacin. 25. Bit je uvijek 0, a preostala 23 bita direktno uzmemo iz niziih 23 bita IP multicast adrese.

Primjer:

Ako je multicast IP adresa 224.0.0.10 onda je multicast MAC adresa 01-00-5E-00-00-0A.

Ako host dobije broadcast frejm koji nije za njega,on kad vidi detalje u broadcast frejmu on ce reci “aha, to je za sve, pa I za mene”, onda ce doci do broadcast IP adrese, I on ce reci “to je za sve,pa I za mene”, a tek kada dodje do segmenta on ce vidjeti da to nije za njega. Ako dobije multicast frejm koji nije za njega, on ce vec odmah na osnovu destinacione multicast adrese moci da zakljuci da nije unutar te multicast grupe I moci ce odmah da odbaci taj frejm. U ovom slucaju ce odraditi mnogo manje procesiranja ako frejm nije za njega, moci ce odmah da ga odbaci, a ako je broadcast frejm, morace da dodje da otpakuje do segmenta pa da ga odbaci.

Broj razlicitih multicast IP adresa : 2^28

Broj razlicitih multicast MAC adresa : 2^23

Vise ima multicast IP adresa nego multicast MAC adresa, pa kad budemo vrsili preslikavanje odnos je 32:1, sto znaci da da se 32 razlicite multicast IP adrese preslikavaju u jednu te istu multicast MAC adresu.

Razlicite multicast grupe se mogu predstavljati istom multicast MAC adresom, pa se treba biti oprezan pri radu sa ovim adresama.

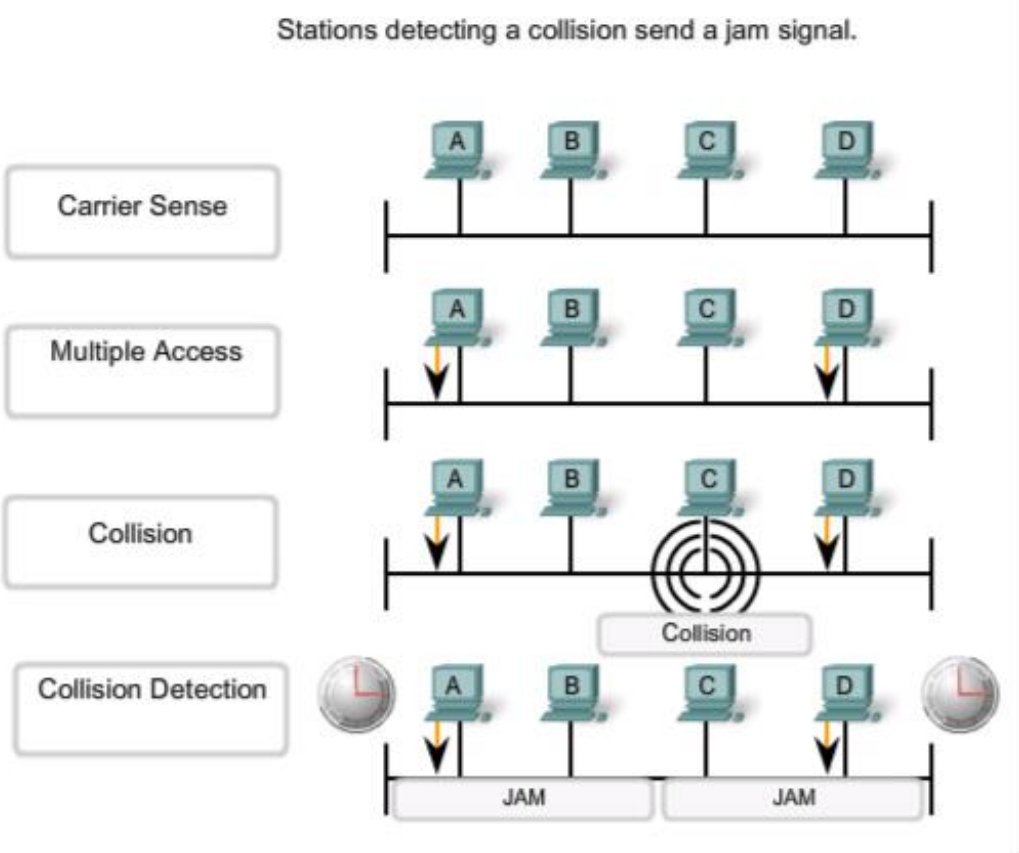
Protokol CSMA/CD = carrier sense multiple access/collison detection

Ethernet koristi protokol CSMA/CD koji je contention-based pristup, to znaci da ako imam nesto da saljem, prvo se provjeri da li je kanal slobodan, ako jeste ja cu da saljem, ako nije, cekam da se kanal oslobodi, pa onda saljem.

Problem moze da nastupi ako dvije stanice istovremeno vide da je kanal Slobodan, onda salju istovremeno, dolaze do sudara signala, I samim tim niko nije uspio da posalje.

CSMA/CD definise sta da radimo kada dodje do kolizije. Zbog toga se ovaj algoritam I zove collison detected, pomocu njega moze samo da se detektuje kolizija, ne moze I da se izbjegne, kolizija je neizbjezna.

Sto je vise racunara, veca je vjerovatnoca da ce dolaziti do kolizije.



A I D su istovremeno vidjel ida je kanal Slobodan I pokusavaju da salju. Ako su oni na nekom HUBu, on to ne moze da procesira,dolazi do sudara tih signala, oba signala se izoblicuju I dolazi do rasta amplitude na tom medijumu. Samim ti, ovi racunari koji su slali mogu da uoce da nesto nije u redu, algoritam samim tim pocinje. Kada ti racunari uoce da to nije u redu, njihov zadatak je da obavijeste sve ostale korisnike na kanalu da nesto nije u redu, odnosno da je doslo do kolizije. Oni to rade tako sto salju JAM signal. JAM signal je neka predefinisana sekvenca jedinica ili nula, kojim se svi koji se nalaze na tom kanalu obavjestavaju da je doslo do kolizije. Kada svi saznaju da je doslo do kolizije, svi se povlace, niko nece da salje jedan odredjen perio. Svaka stanica koja se nalazi na tom kanalu ce izabrati slucajan vremenski period u toku kojeg ce se povuci sa tog kanala. Bitno je da ova vrijednost bude slucajna, jer u suprotnom ako bi bio fiksan vremenski trenutak, kada bi on istekao svi bi ponovo navalili na kanal. Kada istekne izabrani interval racunaru, on ce opet pokusati da salje.

Sam algoritam povlacenje se naziva Random back-off, svi se povlace na slucajan vremenski period. Kada nekom istekne taj timer, on nema nikakav prioritet u odnosu na kanal, nemaju prioritet uredjaj koji je izazvao koliziju, svi jednako konkurisu za dobijanje kanala.

Carrier Sense – svi osluskuju kanal

Multiple Acces – kolizija se desava kada vise uredjaja pristupi kanalu istovremeno

Collison Detection – pomocu algoritma se vrsi detekcija kolizije, ima poseban algoritam kako da se povuku I kako ponovo da pokusaju da pristupe kanalu

Ovaj algoritam se koristi samo u mrezama gdje imamo dijeljeni pristup kanalu. To je danas malo aktuelno, ako imamo switch ovaj algoritam nam nije potreban. Ako imamo HUB, ovaj algoritam se koristi.

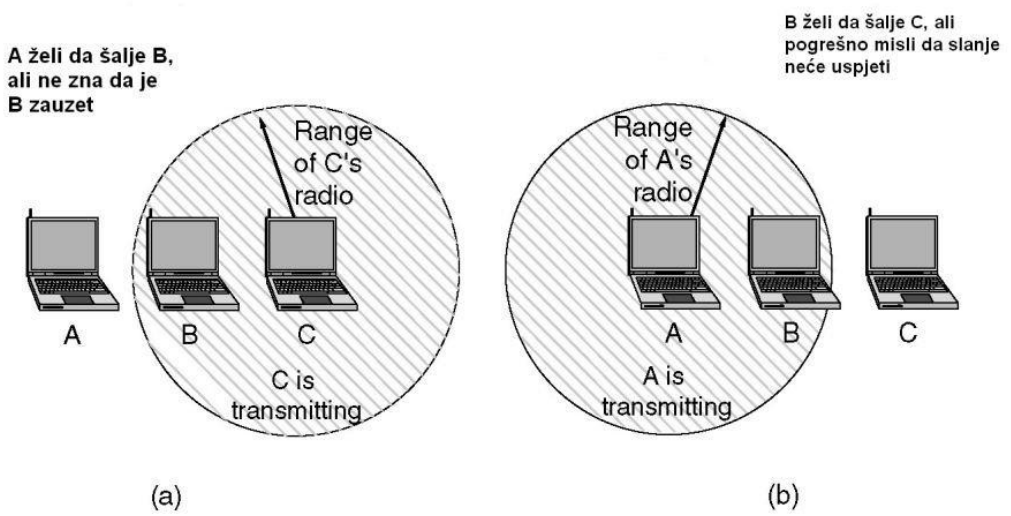
U wireless ne mozemo da koristmo ovaj algoritam, jer u wireless mrezi nemamo jedan kanal kojem svi pristupaju, nego se podaci salju svuda oko nas. Samim tim, ne mogu svi da osluskuju isti kanal.

U wireless mrezama koristimo algoritam CSMA/CA

CA- Collison avoidance = izbjegavanje kolizije

Dva problema koja se mogu pojaviti u wireless mrezama, a ne mogu u Ethernet mrezama:

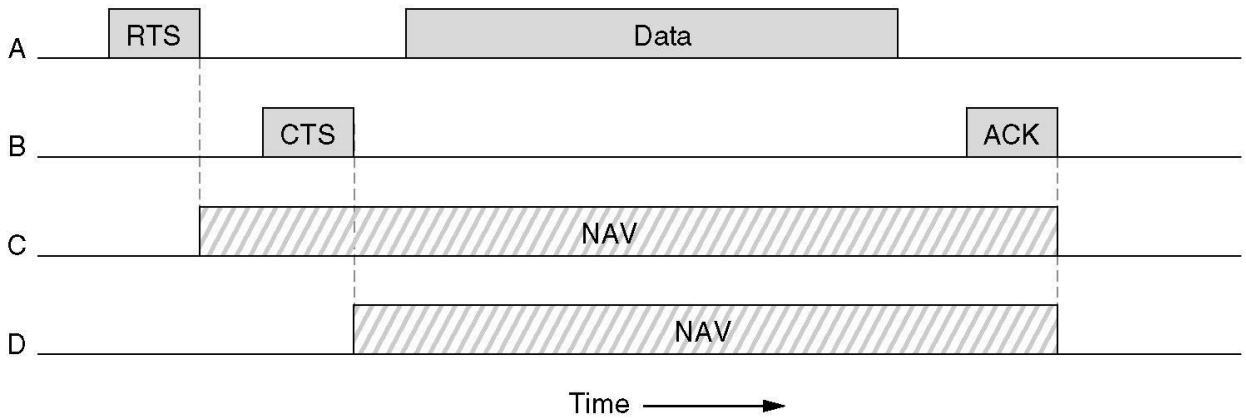
1) Hidden station – skrivena stanica



A u ovom trenutku zeli da salje podatke racunaru B, ali racunar B ne moze da primi te podatke jer vec komunicira sa racunarom C. Racunar A ne moze da zna da C salje podatke, jer nije u rangeu stanice C.

A ce pokusati da salje podatke prema stanici B I doc ice do kolizje, jer B ne moze istovremeno da primi te podatke od A I C. Sada svako ima neki svoj range u kojem radi, samim tim stanica A je van rangea stanica C I stanica C je skrivena stanica za stanicu A I ona ne zna da postoji, ali ona itekako utice na komunikaciju.

2) Izlozena stanica



B zeli da salje podatke prema C, ali to ne radi jer pogresno misli da je zauzeta, a to misli jer je ona u rangeu stanice A, a stanica A uopste ne komunicira sa stanicom B u tom trenutku nego salje podatke prema nekoj stanici D. Tako da je stanica B izlozena stanica stanici A, kao sto je maloprije stanica C bila skrivena stanica stanici A.

Ovi problemi se rjesavaju pomocu algoritma CSMA/CA

Prije bilo kakve komunikacije u wireless mrezama prvo se salju dva mala okvira – RTS (Request to Send = zahtjev za slanje) i CTS (Clear to send = odobrenje za slanje).

Na osnovu lijevog dijagrama

Stanica C prije nego sto pocne komunicirati sa stanicom B prvo salje okvir RTS. Ako stanica B je u tom trenutku slobodna za tu komunikaciju on ace slati okvirt CTS, ali taj okvir dolazi I do stanice C, ali I do stanice A. Samim tim je stanica A obavjestena da stanica B zeli da komunicira sa nekom drugom stanicom, I cim ona primi taj okvir ona aktivira NAV = Network Allocation Vector, koji je tajmer u kojem ona nece pokusavati da komunicira.

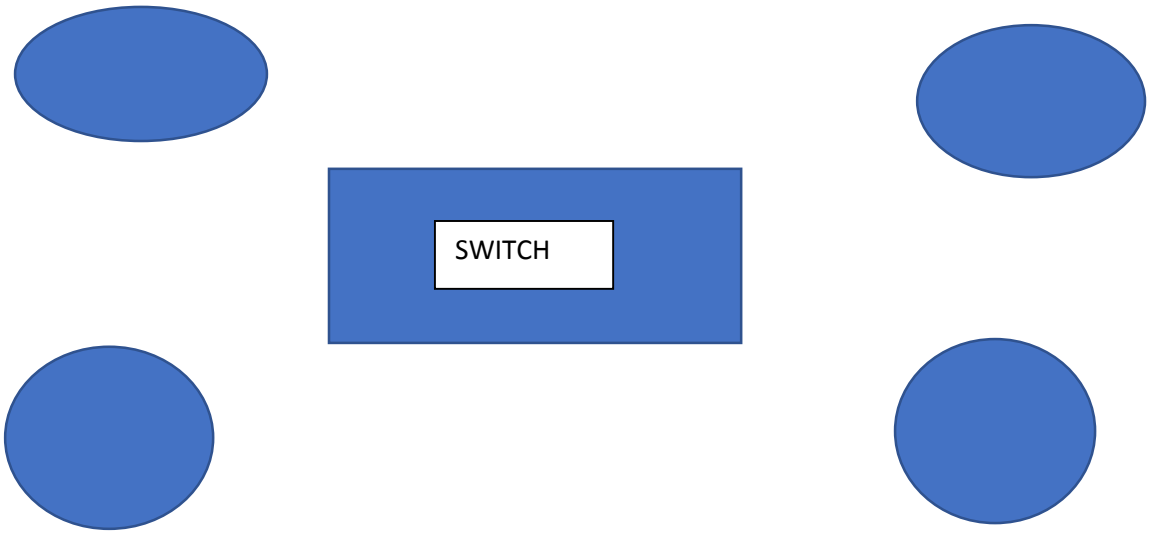
Ovim se rjesava problem, stanica A vise nece biti neupucena da postoji stanica izmeju stanice B I stanice C, na ovaj nacin se izbjegava kolizija, ne moze da se desi, jer ce CTS stici do svih koji su u rangeu stanice B, sto je nama I potrebno.

“Slicno I neki drugi problem”

HUB I SWITCH:

Switch:

Neka imamo Switch na koji su povezana cetiri racunara.



Portovi switcha, mjesta gdje kacimo racunare, su numerisani.

Neka na portu 1 se nalazi racunar sa MAC adresom aa(skraceno), na portu 2 racunar bb, na portu 3 racunar cc, na portu 4 racunar dd.

Na pocetku se aktivira komunikacija, tako da host A zeli da postane neki frejm hostu B. U tom Ethernet frejmu Source MAC = aa, a Destination MAC = bb, oni su u istoj mrezi, tako da mogu to da izvedu. Taj frejm silazi na switch. Switch ima MAC tabelu, to je glavna baza na osnovu koje radi. Ona je krajnje jednostavna baza, jer ima samo dvije kolone. U jednoj koloni se nalazi naziv njegovog porta, a u drugoj koloni MAC adresa uredjaja koji se nalazi na drugom kraju tog porta. Sami portovi Switcha nemaju MAC adrese, pa je to velika razlika u odnosu na ruter. Ruter je radio sa IP adresama, ali su I njegovi interfejsi imali IP adresu. Switch nema MAC adrese. On samo pamti koja je MAC adresa zakacena za koji port.

Inicijalno, MAC tabela Switcha je prazna. Switch dobija neki frejm, I vrsi prosledjivanje iskljucivo na odnosu Destination MAC adrese. Switch uci, puni MAC tabelu, iskljucivo na osnovu Source MAC adrese. Nikako drugacije. Switch dobija taj frejm, I treba nesto da proslijedi, I nesto da nauci. On u ovom slucaju gleda da li ima Destinaton MAC adresu bb u svojoj MAC tabeli. U ovom trenutku nema nista, a ako switch ne zna sta ce sa frejmom, on ga salje na sve portove osim na onaj od kojeg je primio taj frejm. Ruter sa druge strane ako ne zna sta ce sa paketom, on ga odbaci. Switch sa druge strane ako ne zna sta ce sa frejmom proslijedi na sve portove, osim na onaj sa kojeg je frejm stigao.

Switch uci na osnovu Source MAC adresa, I ovdje uci da mu se na portu 1 nalazi adresa aa, I to upisuje u MAC tabelu. Frejm dolazi do sva tri racunara, racunari 3 I 4 vide da frejm nije za njih I odbacuju ga, a racunar 2 prima taj frejm. Kad racunar 2 bude odgovara racunaru 1 on ce napraviti novi frejm sa Source MAC adresom = bb I Destination MAC adresom = aa, I takav frejm silazi na switch. Switch opet radi dvije stvari: prosledjuje I uci.

Kada Switch primi ovakav frejm sa Destination MAC adresom = aa, on gleda da li ima tu adresu u svojoj MAC tabeli. U ovom slucaju ima, I prosledjuje taj frejm iskljucivo prema racunaru aa na portu 1, ne prosledjuje sad frejm svima. Osim toga, u svojoj MAC tabeli, switch upisuje da se na portu 2 nalazi adresa bb.

U nekom realnom vremenu, Switch svoju MAC tabelu popuni za pola minuta, rijetko ce slati na sve portove.

Switch moze da primi dva razlicita frejma sa dva razlicita porta, da ih procesira I proslijedi na odgovarajuce portove, nema gubljenja podataka kao kod HUBa, nema mogucnosti da dodje do kolizije.

Switch je L2 uredjaj I on razumije MAC adrese I na osnovu njih vrsi prosledjivanje frejmova.

HUB je L1 uredjaj, on ne razumije concept frejma, on jedino razumije bite, odnosno signale.

Hub radi tako sto primi signal na jednom portu I jedino sto moze da uradi je da taj signal proslijedi na sve ostale portove. Switch se ponasa kao HUB kad ima praznu MAC tabelu. HUB nema nikakvu memoriju I ne moze da primi dva signala sa dva porta, pa da ih procesira naknadno, pa odmah imamo koliziju. Na HUBu mozemo imati samo jednu komunikaciju u jednom trenutku.

Kolizioni domeni u mrezi: koliko uredjaja u toj mrezi moze istovremeno da salje podatke.

Kod switcha: koliko god imamo aktivnih portova, toliko imamo kolizionih domena(u primjeru je to 4).

Kod HUBa: samo jedan

Ako imamo HUB, svi na HUBu su u jednom domenu, ako imamo Switch, svaki port switcha je zasebni kolizioni domen.

Ako imamo Switch koji radi na 100Mbit/s, tada svaki port ima 100Mbit/s. Kod HUBa se bandwith dijeli, tako da imamo HUB sa 10Mbit/s I imamo tri uredjaja povezana na HUBu, tada svaki uredjaj ima po 3.33 Mbit/s.Ako prikljucimo jos jedan uredjaj, tada svaki uredjaj ima brzinu od 2.5Mbit/s.

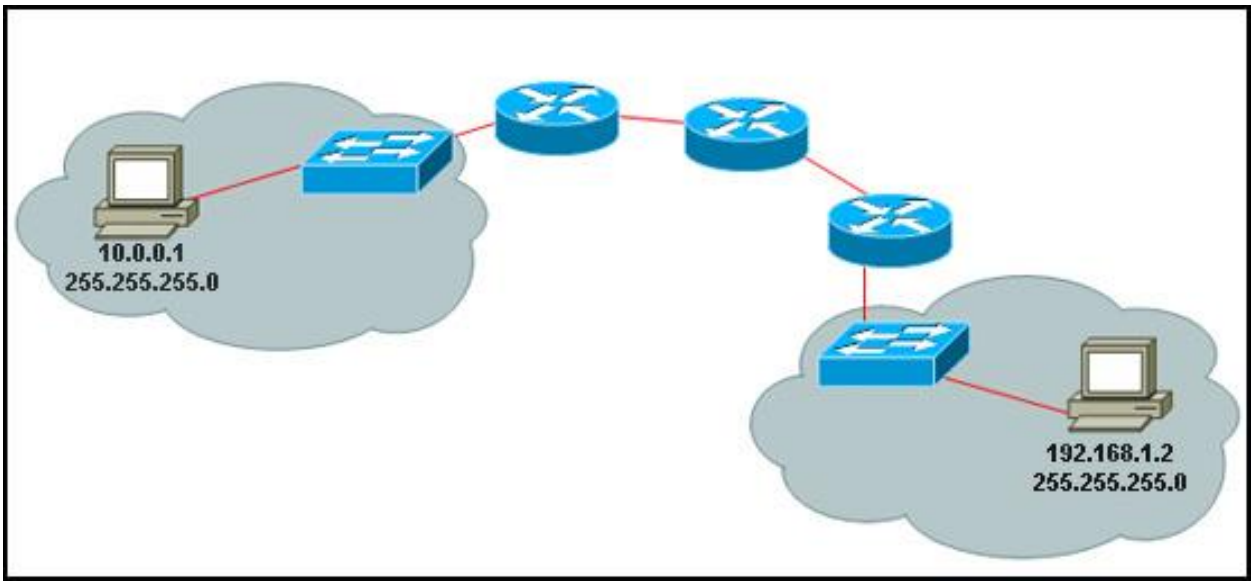
Postoje dva tipa switcheva:

1) Store and forward
2) Cut and through : brzi switchevi zbog kojih se Destination MAC adresa u frejmu nalazi ispred Source MAC adrese.

Switch 2) cim primi prvih sest bajtova frejma, on odmah zna sta ce sa ostalima bajtovima, I odmah Ih salje na taj port gdje je naznaceno. Nece vrsiti nikakvu provjeru, samo prosledjuje frejm dalje.

Store and forward primi cijeli frejm, izvrsi verifikaciju CRC-a, ako je frejm dobar, onda ga salje dalje, ako nije odbacuje ga.

Store and forward su mnogo pouzdaniji.



1 + 1 + 2 + 2 + 0 + 1

- Koliko puta će se vršiti izračunavanje CRC-a u datoj topologiji na slici pri slanju paketa od jednog do drugog hosta? Jedan svič radi u *store-and-forward*, a drugi u *cut-through* režimu. Obrazložiti etape u računanju.

Prvi switch je Store and forward, a drugi je Cut through.

Store and forward(vrsi verifkaciju CRC (zbog toga se vrsi jedno izracunavanje na prvom switch), a Cut through ne vrsi verifikaciju CRC (zbog toga je 0 gore).

Primjeri

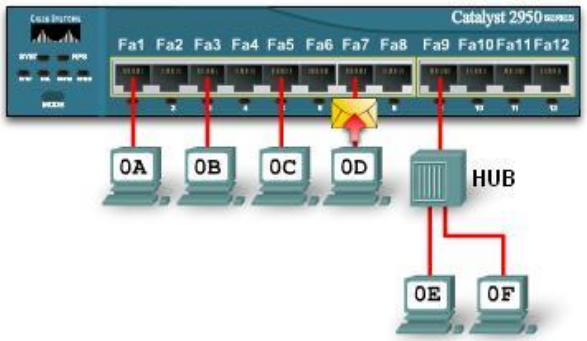
Activity

Determine how the switch forwards a frame based on the Source MAC and Destination MAC addresses and information in the switch MAC table.

Answer the questions below using the information

Preamble	Destination MAC	Source MAC	Length Type	Encapsulated Data	End of frame
	0F	0D			

MAC Table					
Fa1	Fa2	Fa3	Fa4	Fa5	Fa6
0A					
Fa7	Fa8	Fa9	Fa10	Fa11	Fa12
0D		0E 0F			



1. Where will the switch forward the frame?

☐ Fa1 ☐ Fa4 ☐ Fa7 ☐ Fa10
☐ Fa2 ☐ Fa5 ☐ Fa8 ☐ Fa11
☐ Fa3 ☐ Fa6 ☑ Fa9 ☐ Fa12

2. When the switch forwards the frame, which statement(s) are true?

☐ Switch adds the source MAC address to the MAC table.
☐ Frame is a broadcast frame and will be forwarded to all port
☑ Frame is a unicast frame and will be sent to specific port or
☐ Frame is a unicast frame and will be flooded to all ports.
☐ Frame is a unicast frame but it will be dropped at the switch

0D salje na 0F. Switch prima taj frejm, I prvo gleda da li ima 0F u svojoj MAC tabeli. Vidi da ima, I da se nalazi na portu 9, I switch salje samo frejm na port 9. Ne mijenja nista u MAC tabeli, jer je vec znao sta se nalazi na port 9.

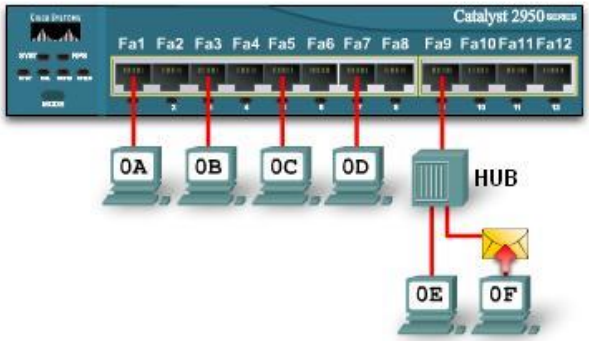
Activity

Determine how the switch forwards a frame based on the Source MAC and Destination MAC addresses and information in the switch MAC table.

Answer the questions below using the information

Preamble	Destination MAC	Source MAC	Length Type	Encapsulated Data	End of frame
	FF	0F			

MAC Table					
Fa1	Fa2	Fa3	Fa4	Fa5	Fa6
		0B			
Fa7	Fa8	Fa9	Fa10	Fa11	Fa12



1. Where will the switch forward the frame?

☑ Fa1 ☐ Fa4 ☑ Fa7 ☐ Fa10
☐ Fa2 ☑ Fa5 ☐ Fa8 ☐ Fa11
☑ Fa3 ☐ Fa6 ☐ Fa9 ☐ Fa12

2. When the switch forwards the frame, which statement(s) are true?

☑ Switch adds the source MAC address to the MAC table.
☑ Frame is a broadcast frame and will be forwarded to all port
☐ Frame is a unicast frame and will be sent to specific port or
☐ Frame is a unicast frame and will be flooded to all ports.
☐ Frame is a unicast frame but it will be dropped at the switch

0F salje na broadcast adresu, tj. Svima u mrezu. Kada taj frejm dodje na HUB, on prosledjuje na oba svoja porta. Taj frejm ce doci I na 0E, ali ce doci I na switch. Switch kada primi broadcast frejm, salje na sve portovoe, osim na onaj sa kojeg je primio taj frejm, odnosno nece ga vratiti nazad, nego salje na portove 7,5,3,1 . Switch ce da nauci da je adresa 0F na portu 9.

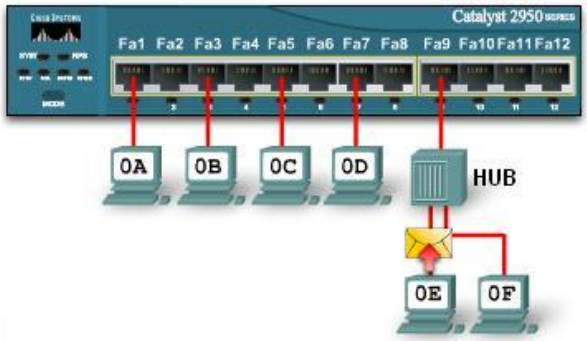
Activity

Determine how the switch forwards a frame based on the Source MAC and Destination MAC addresses and information in the switch MAC table.

Answer the questions below using the information

Preamble	Destination MAC	Source MAC	Length Type	Encapsulated Data	End of frame
	0F	0E			

MAC Table					
Fa1	Fa2	Fa3	Fa4	Fa5	Fa6
		0B			
Fa7	Fa8	Fa9	Fa10	Fa11	Fa12
0D		0F			



1. **Where will the switch forward the frame?**

☐ Fa1 ☐ Fa4 ☐ Fa7 ☐ Fa10
☐ Fa2 ☐ Fa5 ☐ Fa8 ☐ Fa11
☐ Fa3 ☐ Fa6 ☐ Fa9 ☐ Fa12

2. **When the switch forwards the frame, which statement(s) are true?**

☑ Switch adds the source MAC address to the MAC table.
☐ Frame is a broadcast frame and will be forwarded to all port
☐ Frame is a unicast frame and will be sent to specific port or
☐ Frame is a unicast frame and will be flooded to all ports.
☑ Frame is a unicast frame but it will be dropped at the switch

0E salje prema 0F, to dolazi na HUB, I Hub prosledjuje na oba ostala porta, dolazi do switcha I 0F. Switch ce nauci da se 0E nalazi na tom portu, I vise MAC adresa moze biti na jednom portu. Switch treba da frejm proslijedi prema 0F, a pise da je 0F na portu 9, ali to bi znacilo da switch mora da vrati taj frejm nazad odakle je dosao, a on to ne smije, pa se frejm odbacuje od strane switcha u ovom slucaju. To je I logicno, jerk ad ne bi odbacio taj frejm, to bi znacilo da ce 0F dva puta primiti isti frejm, a to ne treba da se desi.

Switch ne moze nikad da nauci multicast ili broadcast adresu, jer ne moze broadcast adresa da bude Source adresa. Switch nikad ne moze da razdvoji broadcast domen, svi uredjaju koji su na switchu su u istom broadcast domenu.

Learning – switch moze da zapamti na kojem portu je koja adrea, process ucenja.

Aging – znaci da switch nece vjecno da drzi neku MAC adresu u tabeli, ona se drzi neki vremenski interval, zavisno od switcha do switcha. Ako se u tom periodu ne desi nikakva komunikacija na tom portu, on brise tu MAC adresu. Port mora da bude aktivan u tom period, da bi se produzio zivot tog zapisa u MAC tabeli.

Flooding – plavljenja, opcija da switch salje na sve portove, osim na onaj sa kojeg je primio frejm, kada nema zapisa u MAC tabeli

Selective forwarding – na osnovu MAC tabele vrsi prosledjivanje na odredjeni port.

Filtering – switch moze da uradi da isfiltrira frejm I da ga odbaci jer je zakljucio da ne moze da ga vrati nazad na port odakle je I dosao.

Kada otkucamo simbolicko ime u pretrazivacu, npr. google.com, pomocu DNS cemo dobiti IP adresu tog sajta, ali postavlja se pitanje kako znati na koju MAC adresu treba proslijediti podatke.

Za to se koristi protokol ARP.

On na osnovu poznate IP adrese na koju salje, nabavlja nepoznatu MAC adresu (L2 adresu).

Ako komuniciramo u svojoj mrezi,

Imamo tri racunara, povezana na switch:

Prvi racunar ima IP 10.1.1.1 I MAC adresu aa, drugi racunar 10.1.1.2 i bb, a treci racunar 10.1.1.3 I cc.

Ako na racunaru aa otkucamo ping 10.1.1.2, odmah smo dobili Destinacionu IP adresu, zna na koju adresu da adresira paket, ali ne zna na koju fizicku adresu da adresira frejm. Tu se aktivirala protokol ARP.

Ako racunar ne zna koja je MAC adresa koja odgovara adresi 10.1.1.2, on automatski aktivira ARP tako sto salje poruku koja se zove ARP request. ARP request je broadcast poruka, dolazi do svih uredjaja u mrezi, I to je vapaj za pomoc, jer ovim se pita da li neko moze da odgovori koja MAC adresa odgovara IP adresi 10.1.1.2. Ako postoji uredjaj sa adresom 10.1.1.2 on ce se javiti i poslace njemu ARP replay, I javice da je on sa IP adresom 10.1.1.2 i da je njegova MAC adresa bb. Nakon ovog host moze da formira frejm sa Source MAC adresom aa I sa MAC adresom bb.

Ovo je drugi veliki protokol koji koristi broadcast poruke.

Svaki racunar ima ARP cache, I tu se nalazi tabela gdje se nalazi IP adresa I MAC adresa, I on ce nakon ovog koraka upisati da adresi 10.1.1.2 odgovara MAC adresa bb. Sledeci put kada bude trebao slati nesto na ovu IP adresu, on ce prov pogledati ARP cache da li ima to preslikavanje I dalje, ako ima, onda ce iskoristiti taj podatak.

Kao I MAC tabela, I ARP cache nece vjecno imati to preslikavanje. Kod Windowsa, ako se zapis u ARP cacheu ne iskoristi u toku dvije minute ponovo, on se brise, a ako se iskoristi, onda se ceka 15 minuta za ponovnu provjeru, ako se ne iskoristi u tom periodu brise se (tako je odradjeno na Windowsu).

Primjer:

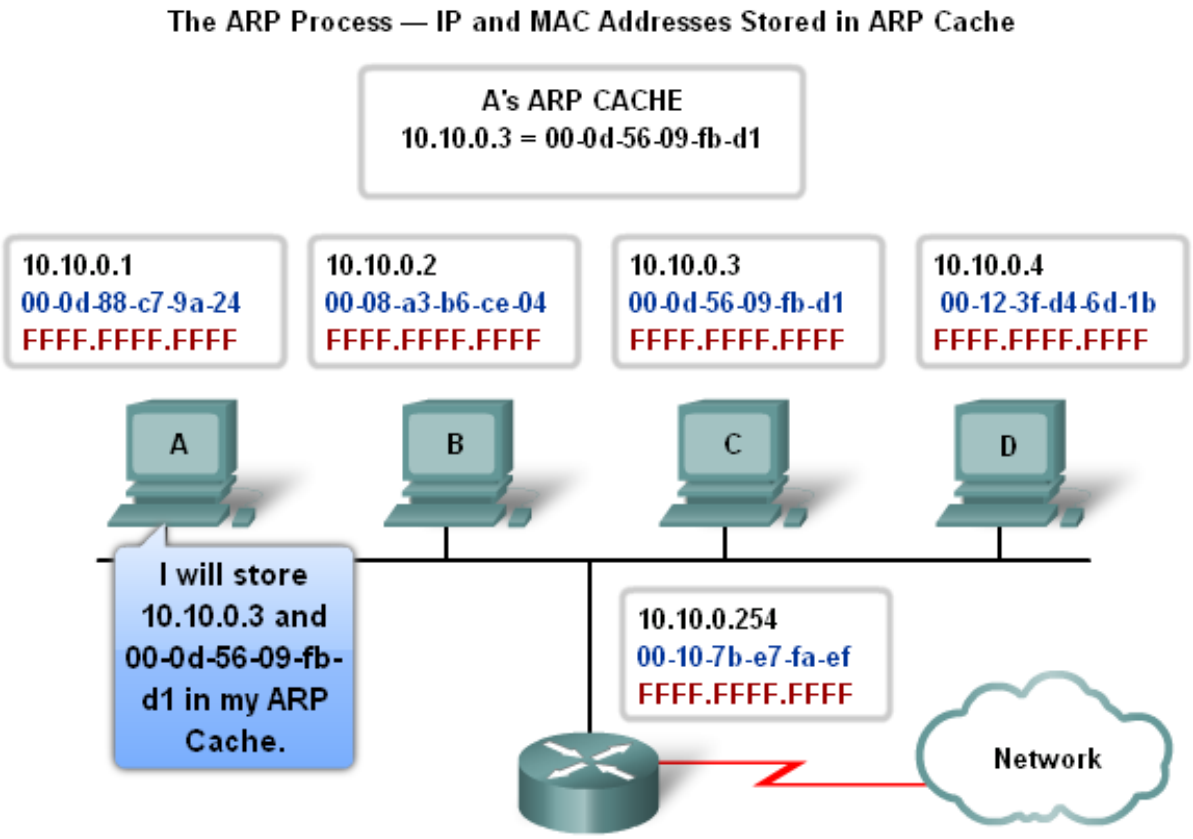
Racunar aa 10.1.1.1

Switch 10.1.1.254 dd ruter internet

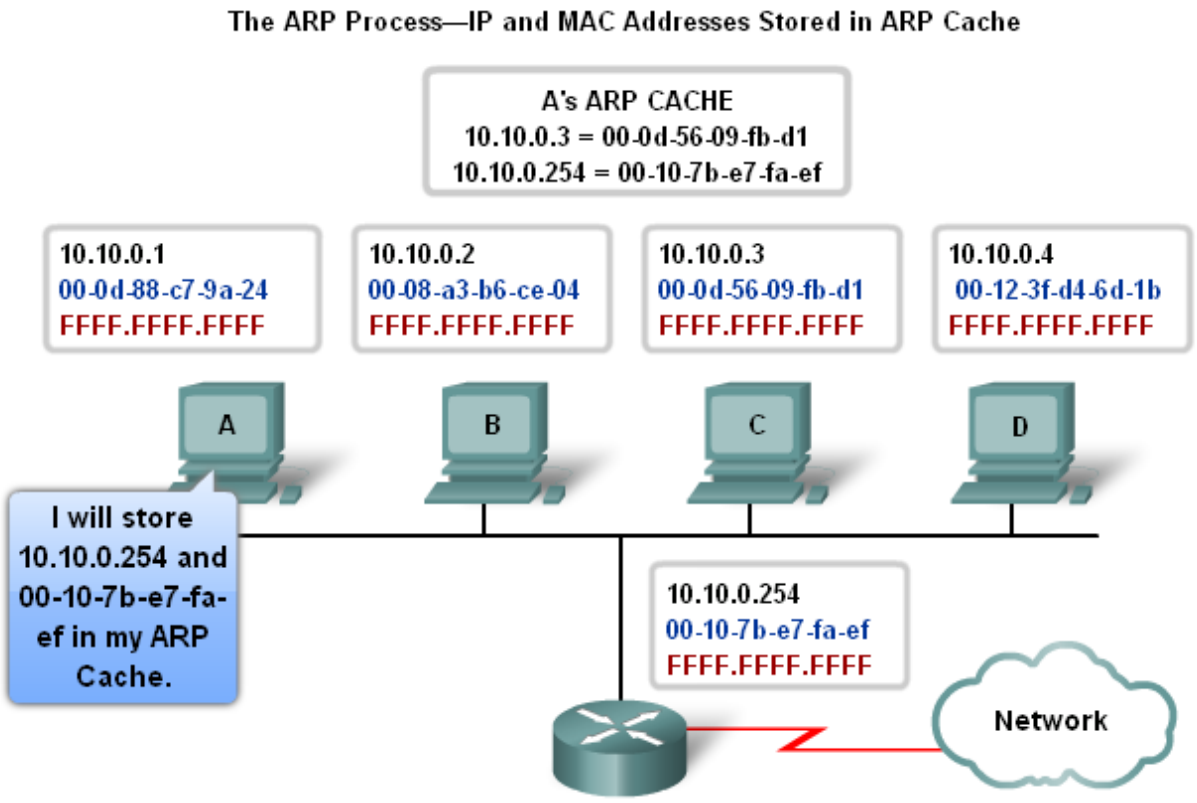
Racunar bb 10.1.1.3

Pingamo google.com sa aa racunara. Host aa zakljucuje da google.com nije u njegovoj mrezi. U ARP requestu host A onda trazi MAC adresu defaultnog gatewaya, jer mu informacija o MAC adresi google.com nista ne znaci, jer nisu na istoj mrezi. U svom ARP cacheu, on upisuje da defaultnom

gatewayu sa IP adresom 10.1.1.254 odgovara MAC adresa dd. U Arp cacheu se mogu nalaziti samo preslikavanja iz njegove mreze, ako ide van mreze, onda mu treba defaultni gateway I bice to preslikavanje. Nikad se ne moze nalaziti preslikavanje za neku adresu koja nije u njegovoj mrezi.



Slanje unutar mreze



Slanje van mreze

PROXY ARP:

Nekad se moze desiti da je neki host pogresno podesen I da greskom zatrazi MAC adresu IP adrese koja nije na njegovoj mrezi. Onda u takvim slucajevima na ruteru mozemo imati konfigurisan protokol koji se zove Proxy ARP. Ruter kada primi takav zahtjev, a primice jer je broadcast poruka, lazirace svoj identitet I reci ce “Ja sam taj, posalji meni taj paket, pa ako ja uspijem nesto da uradim super, ako ne, ionako ne bi proslo”.

Problemu kod ARPa je taj da ako se ruter mogao lazno predstavljati, zasto se I neko drugi ne bi mogao lazno predstavljati. Ako imamo switch, I napadac se prikljuci na taj switch, ako dolazi ARP request gdje se trazi MAC adresa koja odgovara defaultnom gatewayu, napadac moze slagati u ARP reply I dati svoju MAC adresu umjesto adresu defaultnog gatewaya. Posljedica ovog da sav saobracaj koji treba ici van te mreze, ici ce prema napadacu.

Zastita od ovog je kontrola uredjaja koji se prikljucuju u mrezu.

Problem kod ARPa su broadcast poruke, jer mogu ove poruke dobro da zaguse mrezu, pogotovo ako je velika mreza.

Tipovi UTP kablova:

UTP kabl ima 8 zica, odnosno 4 parice. Ako koristimo 100Mbitne kablove, koristimo samo cetiri zice za slanje bita, 4 se uopste ne koriste, ako koristimo gigabitni internet, koristimo osam zica.

Zice 1 I 2 koristimo za slanje podataka, a 3 I 6 za primanje podataka, za 100Mbitne kablove.

Sve uredjaje koji komuniciraju u racunarskim mrezama mozemo podijeliti na: normalne i druge "cudnije",

U prvu kategoriju spadaju racunari I ruteri, a u ovu drugu switchevi I Hubovi. Normalni uredjaj koristi zice 1 I 2 za slanje, a 3 I 6 za primanje.

Druga kategorija koristi 1 I 2 za primanje, a 3 I 6 za slanje.

Zbog toga trebaju dva tipa kablova:

1) Straight-through (strejt)
2) Crossover (kros)

Postoje dva standarda kako da poredamo boje na krajevima zica:

1) T568A

2) T568B

	1	2	3	4	5	6	7	8
T568A	ZB	Z	PB	B	NB	N	SB	S
T568B	NB	N	ZB	P	PB	Z	SB	S

Na parnim pozicijama je puna boja, na neparnim kombinacija boje I bijele bolje.

Straight kabl dobijamo ako na obe strane kabla stavimo A, ili ako na obe strane kabla stavimo B.

Kros ako na jednu stranu stavimo A, a na drugoj strani B.

Straight kablovi se koriste za povezivanje uredjaja razlicitih kategorija.

Kros kablovi se koriste ako se povezuju uredjaji istih kategorija.

Svi novi uredjaji imaju ugradjenu funkciju Auto MDIX, uredjaj moze da prilagodi svoj interfejs kablu koji je u njega uboden.