SVEUČILIŠTE U SPLITU
SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE U SPLITU

ŠIROKOPOJASNE MREŽE
Zbirka zadataka u “Cisco Packet Tracer” programskom alatu

Toni Jončić

PRIJEDIPLOMSKI STRUČNI STUDIJ ELEKTRONIKE

Sadržaj

Popis slika

Popis tablica

Predgovor

Ovaj repetitorium sa zbirkom laboratorijskih vježbi namjenjen je studentima prijediplomskog stručnog studija elektronike za predmet „SEL025 Širokopojasne mreže“.
Pojam „širokopojasne mreže“ uveden je u naš riječnik pojavom ISDN-a koji je pretplatnicima uz samu uslugu telefoniranja nudio i razne dodatne usluge (prikaz broja, prikaz tarifiranja, itd... ali i pristup Internetu) koje su sve bile omogućene kroz jednu vrstu tehnologije. Sama definicija „širokopojasnih mreža“ prvobitno je označavala mreže s brzinom prijenosa podataka većom od 2Mb/s. Napretkom tehnologije te granice su vrlo brzo nadmašene pa se danas „širokopojasnim mrežama“ nazivaju one mreže koje su u stanju krajnjem korisniku pružiti tzv. „*triple play*“ usluge, tj. usluge telefoniranja, praćenja TV programa, te pristupa Internetu velikim brzinama.
Obzirom da našem studiju postoji obvezni kolegiji „Lokalne i pristupne mreže“ i „Računalne mreže“ koji detaljno obrađuje tehnologije pristupnih mreža na nivou prvog i drugog sloja OSI modela, u ovoj skripti naglasak će biti na protokolima trećeg (mrežni) i četvrtog (prijenosni) sloja te vještinama i znanjima potrebnim za konfiguriranje mrežne opreme. Pri odabiru simulacijskog programskog alata za izvođenje laboratorijskih vježbi odabran je „Cisco Packet Tracer“ jer se Cisco operativni sustav (i naredbe koji se koriste za konfiguraciju) najčešće koriste u profesionalnim telekomunikacijskim mrežama.

Autor

1. Upoznavanje s računalnom opremom i aplikacijom „Cisco Packet Tracer“

Više računalnih uređaja međusobno spojenih čine računalnu mrežu - LAN (*Local Area Network*). Danas najrasprostranjenija tehnologija koja se koristi za izradu LAN mreža je Ethernet.

Svako računalo (ili bilo koji drugi uređaj) spojeno na LAN mrežu jedinstveno je određeno **MAC (*Media Access Control*) adresom** koja je trajno zapisana u njegovu mrežnu karticu od strane proizvođača prilikom izrade u tvornici. Svakom proizvođaču mrežnih kartica dodjeljen je pojas MAC adresa koje po redu upisuje u svoje mrežne (Ethernet, WiFi,…) kartice i tako je zajamčeno da ne postoje na svijetu dva sučelja sa istom MAC adresom. Međutim, te mrežne kartice kasnije se prodaju tvornicama računala diljem svijeta, a računala se pak prodaju kupcima diljem svijeta što dovodi do toga da se bilo koja MAC adresa može pojaviti u bilo kojem uređaju u bilo kojem dijelu svijeta. Već iz tog saznanja postaje razvidno da bi bilo nemoguće organizirati globalnu razmjenu paketa koristeći MAC adrese.

Kako bi se omogućila komunikacija i sa drugim računalima u svijetu, neophodno je bilo uvesti globalno struktruiran adresni sustav čije adrese se neće neizbrisivo zapisivati u HW računala već će se po potrebi moći dodjeljivati svakom računalu u ovisnosti u kojem dijelu svijeta se računalo trenutno nalazi. To su takozvane **IP adrese (adresiranje na trećem sloju OSI modela)** koje su strukturirane na način da omogućuju razmjenu IP paketa i sa drugim mrežama, tj. omogućuju nam globalno usmjeravanje prometa. Točno se zna koje IP adrese pripadaju uređajima u Europi, koje uređajima u sjevernoj Americi, koje u južnoj…itd. IP adrese su jedinstvene na svijetu, ali su uredno geografski distribuirane. Nisu vezane za HW računala već su vezane za određenu regiju, određenu državu u toj regiji, za određenog operatera i svaka je na kraju dodjeljena točno nekom pretplatniku. Na taj način znatno je lakše izvršiti globalno usmjeravanje paketa.

Sve računalne aplikacije rade koristeći IP adrese (ne MAC adrese), pa svako računalo mora graditi tablicu u kojoj će povezivati IP i MAC adrese uređaja u svojoj LAN mreži kako bi znalo ispravno adresirati Ethernet okvire. Računala u LAN mreži međusobno razmjenjuju poruke koje im omogućuju da saznaju podatke jedni o drugima (tzv. ARP poruke - razlučivanje MAC i IP adresa).

Kada neko računalo unutar LAN mreže ostvarene Ethernet tehnologijom šalje podatke drugome računalu, podatke ubacuje u Ethernet okvir (*frame*), kao odredišnu adresu okvira postavi MAC adresu odredišnog računala, a potom vrši slanje tog okvira „žicom“ bit po bit.

Unutar najprimitivnijeg LAN-a možemo zamisliti da su sva računala jednostavno spojena na jednu „žicu“. Uređaj koji nam to omogućava naziva se **zvjezdište (*hub)***. Zvjezdište nema nikakvu „pametnu“ funkciju ali na sebi ima veći broj sučelja (*port*) pa nam omogućuje da na jednostavan način, putem konektora, priključimo ili odspojimo uređaje sa zajedničke „žice“ (LAN mreže), te pruža pojačanje snage signala i na taj način omogućuje veću međusobnu udaljenost računala.

U Ethernet načinu rada, sva računala stalno osluškuju „žicu“ (medij) da vide ima li kakvog prometa po njoj. Ukoliko ima nekog slanja podataka, provjeravaju po MAC adresi da li su ti

podaci namjenjeni njima (tada ih prime i obrade) ili su pak za nekog drugog, pa ih samo zanemare.

Ukoliko neko računalo želi nešto „reći“ tj. poslati neke podatke drugom računalu (*unicast*) ili pak odaslati neku *broadcast* poruku namjenjenu svim računalima u mreži, ono prvo mora „osluškivati“ žicu i tek kad se uvjeri da nitko drugi ne „priča“, može poslati svoje poruke. Ukoliko se dogodi da dva računala istovremeno pošalju podatke događa se „**kolizija“**. Tom prilikom signali (poruke) unište jedan drugog, te se slanje mora ponoviti. Ukoliko na mreži ima veliki broj računala to postaje jako veliki problem. Mreža se uspori jer od prevelikog broja „govornika“ više nitko ne može „doći do riječi“.

Kako bi se riješio taj problem prvo se razvojem tehnologije pojavio uređaj zvan **most (*bridge)***, a kasnije i **preklopnik (*switch)***. To su uređaji koji izoliraju kolizijske domene (područja kolizija). Preklopnik (*switch*), isto kao i „*hub*“ uređaj, ima više sučelja za spajanje mrežnih uređaja ali preklopnik **pamti MAC adrese** uređaja priključenih na svako njegovo sučelje (*port*). Kod preklopnika je komunikacija izolirana samo na sučelja kojima su spojeni uređaji koji trenutno komuniciraju. Istovremeno je moguća komunikacija neka druga dva uređaja preko druga dva sučelja istog preklopnika. Preklopnici omogućuju ***full-duplex*** način komunikacije. Kod *full-duplex* načina rada oba uređaja koji komuniciraju mogu slati i primati podatke istovremeno, a da se ne dogodi kolizija. Za međusobno spajanje računala unutar LAN mreže *hub* uređaji se danas više uopće ne koriste, već isključivo preklopnici (*switch).*

Slika 1.1. *Preklopnik (switch)*

Preklopnik ili „Switch“ je relativno jednostavan za uporabu. To je uređaj koji radi na drugom sloju OSI modela tj. prospaja promet sa sučelja na sučelje temeljem MAC adresa.

Zašto kažemo da preklopnik „radi na drugom sloju OSI modela“?

Zato što on prilikom rada obrađuje samo zaglavlje drugog sloja tj. Ethernet zaglavlje te temeljem odredišne MAC adrese prebacuje Ethernet okvire tj. naš „promet“, sa jednog sučelja (dolaznog) na neko drugo sučelje (odredišno). Preklopnik nikada ne gleda što se nalazi u „teretnom“ prostoru Ethernet okvira. Iz tog razloga preklopnici u svome radu izvodi manje operacija u odnosu na usmjerivače te su u stanju brže prosljeđivati promet. U „teretnom prostoru“ Ethernet okvira može se nalaziti paket bilo koje mrežne tehnologije trećeg sloja poput IPv4 paketa, IPv6 paketa, ATM-a,….ili neke buduće mrežne tehnologije koja će se tek pojaviti.

Za jednostavan rad preklopnika nije potrebno posebno konfigurirati priključna sučelja. Preklopnik je dovoljno „pametan“ da kada se kabelom neki uređaj (npr. računalo ili printer) spoji na njega, od njega očita njegovu MAC adresu i zapamti je. Sada kad npr. PC4 želi komunicirati sa PC8, preklopnik (*switch*) prosljeđuje podatke sa PC4 samo na PC8, a ne kao *hub* prema svim računalima u mreži.

Na preklopnik se smiju spajati samo računala unutar **iste IP mreže** (IP mreža je određena IP adresama). Preklopnik u potpunosti eliminira koliziju u mreži, ali ne može kontrolirati *broadcast* poruke. Na preklopniku možemo kreirati VLAN-ove (podijeliti spojene uređaje u grupe) i tako smanjiti probleme *broadcasta*. *Multilayer switch* – najskuplja vrsta preklopnika, može kontrolirati i kolizije i broadcast.

Usmjerivač ili „Router“ – usmjerava mrežni promet između **različitih IP mreža**. Radi na trećem sloju OSI modela tj. usmjeravanje paketa vrši temeljem IP adresa. Svako njegovo ulazno sučelje(*interface*) moramo mi konfigurirati (tj. upisati mu IP adresu i ostale parametre) i svako njegovo sučelje pripada drugoj IP mreži. Ukoliko neki PC iz jedne mreže želi komunicirati sa uređajem iz druge mreže, *router* zna na kojem se njegovom sučelju (*interface*) nalazi najbolji put prema toj drugoj mreži i prosljeđuje pakete samo prema toj mreži. Usmjerivač ne radi usmjeravanje paketa na razini IP adresa uređaja, već mreža.

Svaki usmjerivač zna koje mreže su direktno spojene na njegova sučelja jer prilikom konfiguracije svakog sučelja moramo upisati njegovu IP adresu i mrežnu masku, pa je samim time određena i pripadnost IP mreži. Kako bi saznao najbolje puteve do ostalih (udaljenih) mreža usmjerivač gradi i obnavlja tzv. **tablicu usmjeravanja** (***routing table***). Što je to? Usmjerivači osim što prosljeđuju promet među mrežama koje direktno povezuju, također stalno međusobno komuniciraju sa ostalim usmjerivačima u mreži kako bi saznali gdje proslijediti informaciju za bilo koju mrežu na svijetu. Ta komunikacija između usmjerivača vrši se pomoću tzv. **usmjerivačkih protokola.** Međusobnom razmjenom tih poruka usmjerivači saznaju gdje se nalaze ostale (udaljene) mreže, te računaju koji je „najbolji put“ do neke udaljene mreže. Informaciju o tom „najboljem putu“ spremaju u svoju tablicu usmjeravanja kako bi ,ukoliko do njih stigne paket namjenjen nekoj od tih udaljenih mreža, znali kroz koje svoje sučelje ga trebaju proslijediti dalje.

Zašto kažemo da usmjerivači (*routers*) „rade na trećem sloju OSI modela“?

Usmjerivači imaju sučelja koja mogu biti ostvarena bilo kojom tehnologijom drugog sloja. Često su ta sučelja ostvarena različitim tehnologijama npr. usmjerivač može imati nekoliko žičnih Ethernet sučelja, nekoliko optičkih Ethernet sučelja , te Wi-Fi sučelje. Bilo koje od navedenih sučelja po primitku okvira drugog sloja, skida taj okvir te „teret“, odnosno u našem slučaju IP paket (PDU trećeg sloj OSI modela), šalje na obradu u „mozak“ (CPU) našeg usmjerivača. Usmjerivač pregledava odredišnu IP adresu (adresu trećeg sloja OSI modela) i temeljem nje donosi odluku kuda će proslijediti taj paket. Sve odluke o prosljeđivanju koje donosi usmjerivač bazirane su na adresama trećeg sloja!

Računalo (PC)

Računala su naši korisnički uređaji na kojima su instalirane razne aplikacije koje nam pomažu u radu. Ukoliko želimo da naše računalo bude spojeno sa ostalim računalima i sa ostalim mrežama, potrebno je:

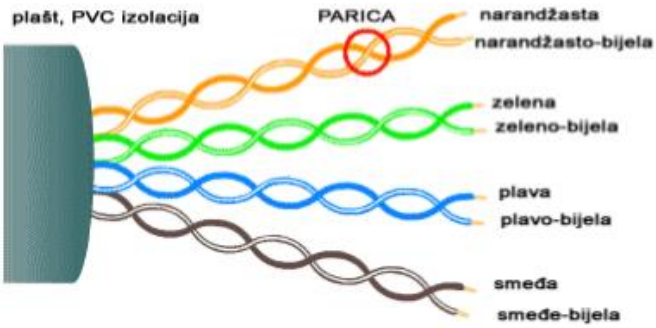
1. upisati njegovu IP adresu,
2. mrežnu masku i
3. IP adresu sučelja usmjerivača (*gateway*) kojemu će računalo poslati IP pakete ukoliko treba komunicirati sa nekim uređajem van vlastite IP mreže.

Postavljanje ovih parametara može se raditi ručno ili automatski putem DHCP protokola.

Uređaji se na mrežu spajaju omeđenim medijem (UTP kabel, koaksijalni kabel, optički kabel) ili bežično (radio vezom). Koaksijalni kabel danas se više gotovo uopće ne koristi. Najčešće se koristi UTP kabel (eng. *Unshielded Twisted-Pair Cable)*, a sve češće su u upotrebi i optički kablovi.

Povezivanje uređaja UTP kabelom (*Straight-trough* i *Crossover* načinom)

UTP (*Unshielded Twisted-Pair Cable*) sastoji se od **4 parice (8 žica**). Za izradu poveznog kabela potreban je muški konektor oznake **RJ-45** i specijalan alat za završno "krimpanje" konektora sa kablom.

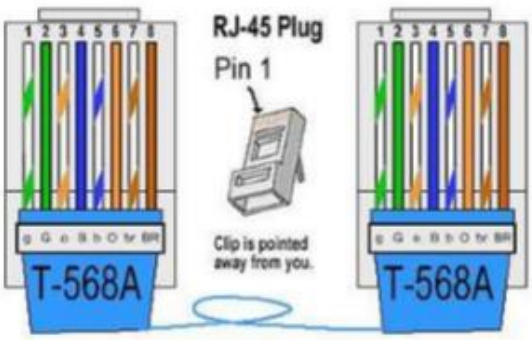


Slika 1.2.*UTP kabel*

Svaka parica u kabelu označena je drugačijom bojom kako bismo ih mogli razlikovati. Postoje dva načina kako se izrađuju spojni kablovi (*Straight-trough* i *Crossover)*, a koristili su se za različite namjene. Današnje mrežne kartice dovoljno su pametne da će i jedna i druga varijanta izrade spojnog kabela omogućiti vezu između uređaja, ali na tržištu još postoje i stariji uređaji pa je potrebno slijediti pravila.

Straight-trough ili *Patch Cable* (Ravni kabel)

Koristi se kod povezivanja 2 različita uređaja u mreži. Poredak žica prikazan je na slici 1.3. (lijeva strana predstavlja jedan kraj kabela a desna drugi kraj).



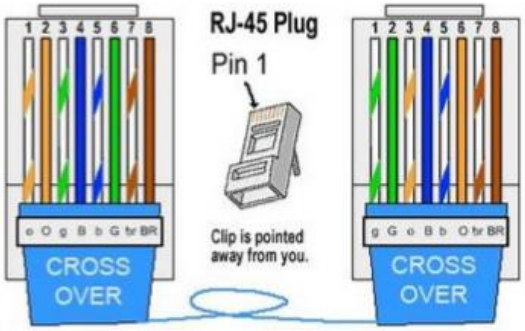
Slika 1.3. *Patch kabel*

Patch kabel koristimo ukoliko želimo povezati sljedeće uređaje:

- Preklopnik i usmjerivač
- Preklopnik i računalo
- Hub i računalo

***Crossover Cable* (Križni kabel)**

Obično se koriste za povezivanje 2 ista tipa uređaja (npr. PC i PC). Poredak žica prikazan je na slici 1.4 (lijeva strana je jedan kraj kabla a desna drugi kraj kabla).



Slika 1.4. *Crossover kabel*

Crossover kabel koristimo ukoliko želimo povezati sljedeće uređaje:

- Računalo i računalo
- Preklopnik i preklopnik
- Preklopnik i *hub*
- *Hub* i *hub*
- Usmjerivač i usmjerivač
- Usmjerivač i računalo (PC)

Načini povezivanja sa preklopnikom ili usmjerivačem u svrhu njihove konfiguracije

Postoje 3 načina spajanja sa mrežnim uređajima (preklopnikom i usmjerivačem):

- ✓ Računalom koje se nalazi u istoj mreži korištenjem ***management* IP adrese.** *Management* adresu potrebno je unijeti u web preglednik na našem računalu, te se na taj način aktivira grafičko konfiguracijsko sučelje (GUI- *Graphical User Interface*) uređaja. Fizičko spajanje našeg računala i uređaja kojeg želimo konfigurirati vrši se Ethernet kabelom. Ukoliko je uređaj već u radu u lokalnoj mreži, možemo mu pristupiti na isti način sa bilo kojeg računala te mreže.
- ✓ Spajanjem računala (serijsko sučelje) sa uređajem putem ***console port*** sučelja. Potreban poseban kabel i SW za uspostavi i kontrolu veze (npr. aplikacija *Putty*)
- ✓ **Telnet** – daljinsko spajanje na uređaj kroz računalnu mrežu. Telnet vezu aktiviramo iz *Command Prompt* grafičkog sučelja našeg računala. Telnet načinom spajamo se isključivo na opremu koja je već konfigurirana i u radu jer je prethodno neophodno postaviti pristupne adrese i lozinke.

Načini spajanja detaljno su opisani u nastavku.

Vrste memorije na uređajima i njihova namjena:

FLASH (trajna memorija) – Ovdje je spremljen operacijski sustav npr. IOS

NVRAM (trajna memorija) – Ovdje su spremljeni konfiguracijski podaci(startup-config file)

ROM (trajna i nepromjenjiva memorija) – Ovdje je spremljen BIOS, POST, i ROMMON.

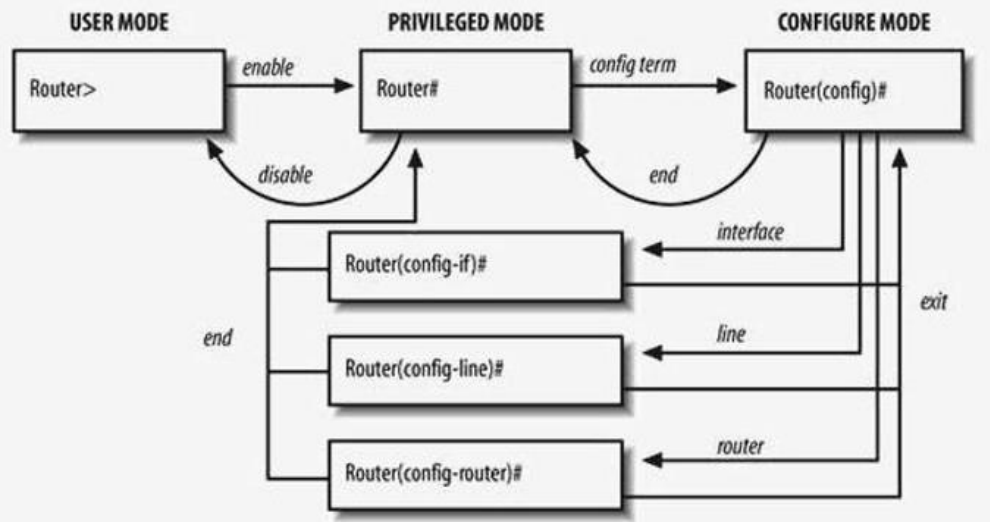
RAM (privremena memorija) – Operacijski sustav i konfiguracijski *file* nekog mrežnog uređaja ovdje se učitavaju prilikom uključenja uređaja iz FLASH i NVRAM memorije. Odavde se i izvršavaju što uređaju pruža veliku brzinu rada. Tablica usmjeravanja također se izvršava iz RAM memorije. Kada napravimo promjene u konfiguraciji one se odmah izvrše u RAM memoriji, ali ukoliko ih želimo napraviti trajnima, moramo ih pohraniti (snimiti) u NVRAM (startup-config) trajnu memoriju.

Konfiguriranje preklopnika i usmjerivača iz CLI (*Command Line Interface*) sučelja

Većina današnjih mrežnih uređaja ima instaliran SW za grafičko konfiguracijsko sučelje (GUI) koje donekle olakšava i ubrzava konfiguraciju samih uređaja. Međutim kroz konfiguracijska sučelja često ne možete konfigurirati baš sve. Također, svaki proizvođač uređaja ima drugačiji GUI. Neki proizvođači opreme uopće nemaju instaliran GUI, već ga je potrebno naknadno instalirati uz nadoplatu. Iz svih gore navedenih razloga, za konfiguraciju uređaja na laboratorijskim vježbama koristiti ćemo se isključivo tekstualnim naredbama iz CLI sučelja. Obzirom da ćemo većinu vježbi raditi u „*Cisco Packet Tracer*“ programu, koristiti ćemo naredbe Cisco operacijskog sustava, koji je ionako najrašireniji na tržištu.

Uređaji (*switch, router*) imaju tri konfiguracijska moda tj. tri razine pristupa koji nam omogućavaju konfiguriranje samih uređaja. To su:

- ✓ *User* mod (ograničeni mod) - samo ograničene konfiguracije.
- ✓ *Privileged* mod– ne može se mjenjati ništa (koristi Admin)
- ✓ *Configuration* mod – konfiguracija uređaja se vrši iz ovog pristupnog moda



Slika 1.5. *Naredbe koje se koriste za prebacivanje iz jednoga moda u drugi.*

„?“ nam služi kao pomoć. U bilo kojem trenutku možemo pogledati koje se naredbe mogu koristiti a one ovise o modu u kojem se nalazimo.

„**Enable**“ naredba koristi se za prijelaz iz *user* moda u *privileged* mod

```
Switch>                          - Nalazimo se u user modu. Lako ga je prepoznati po znaku „>“
Switch>enable                    - Naredbom „enable“ prelazimo privileged mod
Switch#                          - Nalazimo se u privileged modu. Prepoznajemo ga po znaku „#“
Switch#configure terminal        - ovom naredbom prelazimo u konfiguracijski mod
Switch(config)#                  - Nalazimo se u konfiguracijskom modu. „(config) #“.
```

Naredba “exit” vraća nas u prijašnji mod.

```
Switch>enable
Switch#configure terminal
Switch(config)#exit
Switch>
```

Popis naredbi koje su dostupne unutar pojedinog moda

User mode

```
Router>?
Exec commands:
<1-99>          Session number to resume
connect         Open a terminal connection
disable         Turn off privileged commands
disconnect      Disconnect an existing network connection
enable          Turn on privileged commands
exit            Exit from the EXEC
logout          Exit from the EXEC
ping            Send echo messages
resume          Resume an active network connection
show            Show running system information
ssh             Open a secure shell client connection
telnet          Open a telnet connection
terminal        Set terminal line parameters
traceroute      Trace route to destination
Router>
```

Privileged mode

```
Router#?
Exec commands:
<1-99>          Session number to resume
auto            Exec level Automation
clear           Reset functions
clock           Manage the system clock
configure       Enter configuration mode
connect         Open a terminal connection
copy            Copy from one file to another
debug           Debugging functions (see also 'undebug')
delete          Delete a file
dir             List files on a filesystem
disable         Turn off privileged commands
disconnect      Disconnect an existing network connection
enable          Turn on privileged commands
erase           Erase a filesystem
exit            Exit from the EXEC
logout          Exit from the EXEC
mkdir           Create new directory
more            Display the contents of a file
no              Disable debugging informations
ping            Send echo messages
reload          Halt and perform a cold restart
resume          Resume an active network connection
rmdir           Remove existing directory
send            Send a message to other tty lines
setup           Run the SETUP command facility
```

```
show            Show running system information
ssh             Open a secure shell client connection
telnet          Open a telnet connection
terminal        Set terminal line parameters
traceroute      Trace route to destination
undebug         Disable debugging functions (see also 'debug')
vlan            Configure VLAN parameters
write           Write running configuration to memory, network, or terminal
Router#
```

Configuration mode

```
Router(config)#?
Configure commands:
aaa                     Authentication, Authorization and Accounting.
access-list             Add an access list entry
banner                  Define a login banner
bba-group               Configure BBA Group
boot                    Modify system boot parameters
cdp                     Global CDP configuration subcommands
class-map               Configure Class Map
clock                   Configure time-of-day clock
config-register         Define the configuration register
crypto                  Encryption module
default                 Set a command to its defaults
do                      To run exec commands in config mode
dot11                   IEEE 802.11 config commands
enable                  Modify enable password parameters
end                     Exit from configure mode
exit                    Exit from configure mode
flow                    Global Flow configuration subcommands
hostname                Set system's network name
interface               Select an interface to configure
ip                      Global IP configuration subcommands
ipv6                    Global IPv6 configuration commands
```

SW konfiguracija

Konfiguracije koje je moguće izvršiti na SW:

- Hostname
- Negating commands
- Console password
- Telnet password
- Enable password
- Management ip address
- Default gateway
- Shutdown
- Logon banner
- Saving configuration

```
Switch(config)#hostname SW      -Promjena imena
SW(config)#no hostname          -Naredbom „no“ brišemo krivo upisanu naredbu
Switch(config)#
Switch(config)#hostname SW1
```

```
SW1(config)#
```

Lozinke (*Passwords*) na uređaju

Ukoliko smo mi administratori nekog mrežnog uređaja ne želimo dozvoliti da neovlaštene osobe imaju pristup uređaju. Kako bi zaštitili pristup uređaju možemo postaviti lozinke. Postoje 3 različite lozinke koje možemo postaviti na uređaju, svaka štiti jedan od gore navedenih načina pristupa uređaju. To su: *console, telnet i enable password*.

Postavljanje *Console password*

Svi cisco uređaji imaju samo 1 konzolno sučelje (*console port*).

```
SW1#enable
SW1#configure terminal
SW1(config)#line console 0
SW1(config-line)#password 123
SW1(config-line)#login   - bez “login” naredbe naš password će se pohraniti u konfiguraciju uređaja ali
                           neće biti zatražen pri pristupu; “login” naredba zahtjeva upis lozinke da bi se
                           pristupilo tom sučelju.
SW1(config-line)#end
SW1#
SW1#show running-config - ova naredba prikazuje trenutnu konfiguraciju uređaja (iz privileged moda)
            !
            !
            line con 0
            password 123
            login
            !
            line aux 0
            !
            line vty 0 4
            login
            !
```

Postavljanje *Telnet password*

Telnet password nam omogućuje upravljanje našim uređajem sa udaljenosti.

```
SW1(config)#line vty 0 15      - “0 15” znači da se 16 ljudi može istovremeno „telnetirati“ na taj
                                 uređaj
SW1(config-line)#password 1234
SW1(config-line)#login         - kod postavljanja telnet lozinke čak i da zaboravimo upisati ovu
                                 naredbu Cisco uređaj će je sam upisati iz sigurnosnih razloga (bez
                                 nje bi se bilo tko mogao telnetirati na uređaj i raditi što želi)
SW1(config-line)#
SW1#show running-config        - primjetit ćemo da je “line vty” razbijen u 2 djela.
            !
            line vty 0 4
            password 1234
            login
            line vty 5 15
            password 1234
            login
            !
```

Razlog tome je što su stariji Cisco uređaji uvijek imali 5 telnet sučelja koji su išli 0-4. U novijim uređajima dozvoljeno je da se na uređaj može „*telnetirati*“ više od 5 osoba.

To je odvojeno u 2 djela ukoliko želimo kopirati konfiguraciju sa novog uređaja na stari. Stariji uređaji će zanemariti line vty 5 15 jer oni ne znaju za mogućnost telentiranja 16 uređaja istovremeno.

NAPOMENA:

Svakom korisniku je moguće dodijeliti vlastiti *password* tj. ne mora za svaku konekciju biti ista lozinka, može ih biti 16 različitih. Međutim kad se idemo „telnetirati“ sa udaljenog računala na neki uređaj, nikad ne znamo koji *vty line* ćemo dobiti tako da ne bi znali koja lozinka je potrebna da bi se logirali.

Postavljanje *Enable password*

```
SW1(config)#enable password 12345
SW1#show running-config
            version 15.1
            no service timestamps log datetime msec
            no service timestamps debug datetime msec
            no service password-encryption
            !
            hostname Router
            !
            enable password 12345
            !
```

Ovakva konfiguracija nije dobra jer svatko tko može pročitati konfiguraciju, može vidjeti i *enable password*. Zbog toga koristimo sljedeću naredbu:

```
SW1(config)#enable secret 1234567      - enkriptirani enable password
SW1(config)#exit
SW1#show running-config
            no service password-encryption
            !
            hostname Router
            !
            !
            !
            enable secret 5 $1$mERr$iHGIpzTedTCk9bQ93Wry30
            enable password 12345
            !

SW1>
SW1>enable
Password: 12345         - neće nam dozvoliti da se logiramo sa ovom lozinkom
Password: 1234567       - moramo se logirati sa ovom jer Cisco uređaj ovu lozinku vidi kao sigurnu
SW1#
```

Zašto uopće postoji mogućnost zapisa 2 lozinke (*password*)?

Zato što su stari uređaji poznavali samo naredbu „*enable password“* i nisu poznavali „*enable secret“*. Tako da ako neko ide kopirati konfiguraciju sa novog uređaja na stari kopirat će „*enable password (12345)“*.

Zaključak: ukoliko kopiramo konfiguraciju sa novog na stari uređaj, kopirati će se sve starije naredbe. Novije naredbe stari uređaji ne podržavaju i njih će zanemariti. Ukoliko želimo maknuti *enable password* trebamo samo upisati:

```
SW1(config)#no enable password
```

i u konfiguraciji će ostati samo *enable secret password* .

Password **enkripcija**

U ovom trenutku naredbom `SW1#show running-config` možemo vidjeti da *console* i *telnet* lozinke nisu enkriptirane.

```
!
line con 0
password 123
login
!
line aux 0
!
line vty 0 4
password 1234
login
line vty 5 15
password 1234
login
!
```

Enkripciju svih lozinki radimo naredbom:

```
SW1(config)#service password-encryption
```

Naredbom `SW1#show running-config` možemo vidjeti da su sada *console* i *telnet* lozinke enkriptirane.

```
!
line con 0
password 7 08701E1D
login
!
line aux 0
!
line vty 0 4
password 7 08701E1D5D
login
line vty 5 15
password 7 08701E1D5D
login
!
```

Management **(VLAN) IP address**

Omogućava nam pristup preklopniku sa bilo kojeg lokalnog uređaja (koji se nalazi u LAN-u), ali nije moguće pristupit sa uređaja koji nije u istoj mreži (*subnet*).

Naredbom ***„sh ip int brief“*** vidimo sva sučelja (*port*) koji postoje na našem preklopniku. *Status* se odnosi na prvi sloj (L1) OSI modela, a *Protocol* na drugi (L2).

```
Switch>sh ip int brief
```

```
Interface              IP-Address      OK?     Method  Status  Protocol
FastEthernet0/1        unassigned      YES     manual  down    down
FastEthernet0/2        unassigned      YES     manual  down    down
FastEthernet0/3        unassigned      YES     manual  down    down
FastEthernet0/4        unassigned      YES     manual  down    down
!
```

Naredbom „***show vlan***“ vidimo da sva sučelja po početnoj konfiguraciji (*default*) pripadaju vlan-u 1. vlan 1002-1005 već postojeći vlan-ovi, ali nisu podržani (*unsupported*).

```
SW1#show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                Gig0/1, Gig0/2
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        0      0
1002 fddi  101002     1500  -      -      -        -    -        0      0
1003 tr    101003     1500  -      -      -        -    -        0      0
1004 fdnet 101004     1500  -      -      -        ieee -        0      0
1005 trnet 101005     1500  -      -      -        ibm  -        0      0

Remote SPAN VLANs
------------------------------------------------------------------------------


Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------
SW1#
```

Slika 1.6. *Prikaz VLAN konfiguracije*

Ukoliko želimo vlan-u 1 dodijeliti ***managment ip adresu*** moramo izvršiti slijedeće:

```
SW1(config)#interface vlan 1
SW1(config-if)#ip address 10.1.1.10 255.255.255.0
SW1(config-if)#exit
SW1#show ip interface brief
!
GigabitEthernet0/2     unassigned YES manual  down                    down
Vlan1                  10.1.1.10  YES manual  administratively down  down
Switch#


SW1#show running-config

            !
            interface Vlan1
            ip address 10.1.1.10 255.255.255.0
            shutdown
            !
```

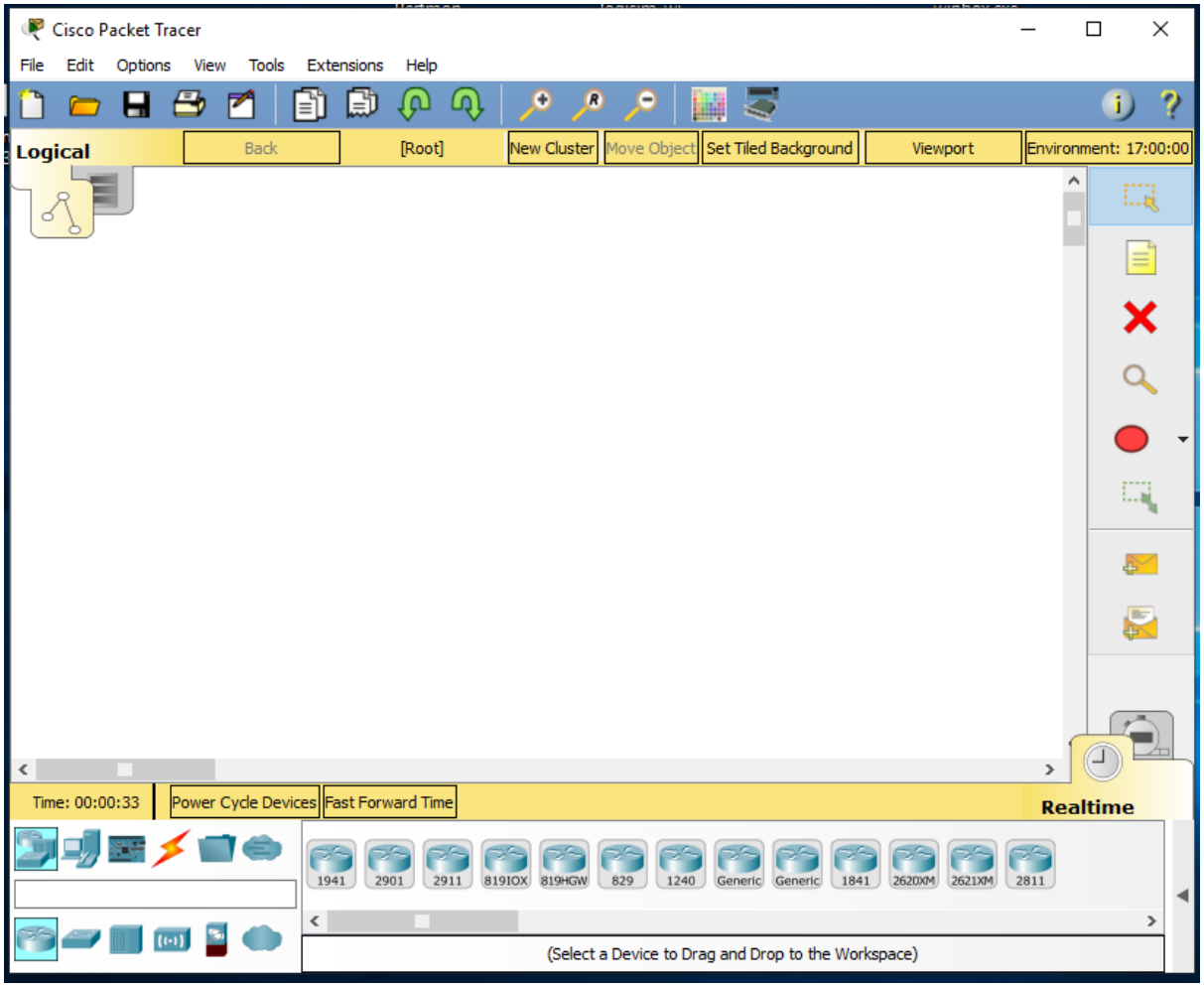
Vidimo da je to sučelje **isključeno** (***shutdown)***. Sva sučelja su tvorničkim postavkama postavljena na isključeno stanje, pa ih uvijek moramo, nakon same konfiguracije, uključiti koristeći naredbu „***no shutdown“***.

```
SW1(config)#interface vlan 1
SW1(config-if)#no shutdown
SW1# show ip interface brief
!
GigabitEthernet0/2     unassigned      YES     manual  down    down
Vlan1                  10.1.1.10       YES     manual  up      down
Switch#
```

Uvod u Cisco Packet Tracer

Cisco Packet Tracer je računalna aplikacija koja nam omogućava virtualno kreiranje različitih računalnih mreža na našem računalu, te konfiguracije i testiranje tih mreža.

Nakon što pokrenete program vidjeti ćete prikaz glavnog zaslona



Slika 1.7. *Izgled glavnog prozora Cisco Packet Tracer aplikacije*

Organizacija PacketTracera:

1. Logički Radni Prostor (*LogicalWorkspace*)
2. Fizički Radni Prostor (*PhysicalWorkspace*)

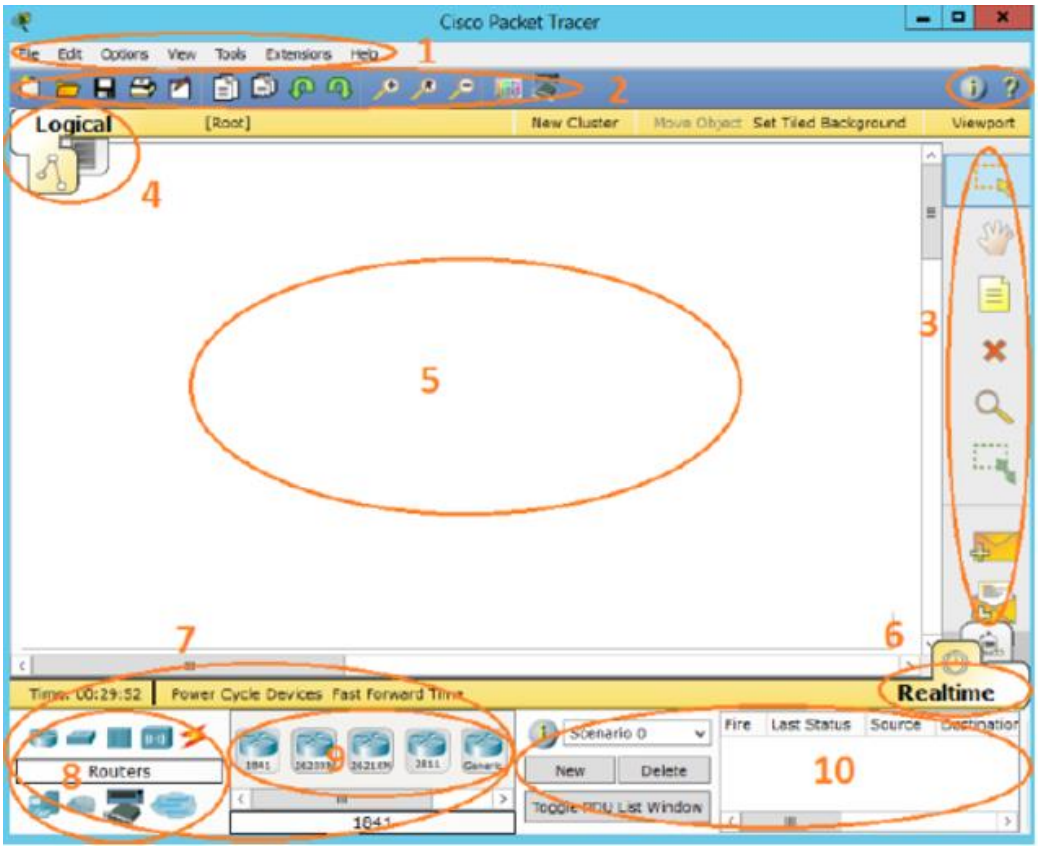
PacketTracer također ima i dva načina rada:

1. Rad u stvarnom vremenu (*Real Time Mode*)
2. Simulacijski način rada (*Simulation Mode*)

Preporuke za rad sa Cisco PacketTracer programom:

- Kada otvorite CPT, naći ćete se logičnom radnom prostoru (*LogicalWorkspace*) u radu u stvarnom vremenu (*Real Time Mode*).
- Ovdje možete graditi svoju vlastitu mrežu i vidjeti rad u stvarnom vremenu
- Možete se prebaciti na simulacijski mod rada (*Simulation Mode*) za pokretanje kontroliranih mrežnih scenarija.
- Možete se prebaciti na fizički radni prostor (*PhysicalWorkspace*) i tako organizirati fizičke aspekte, kao što su lokacije vlastith uređaja.
- Nije moguće pokrenuti svoju mrežu, dok ste u fizičkom radnom prostoru. Nakon što ste završili rad u fizičkom radnom prostoru (*PhysicalWorkspace*) potrebno je vratiti se na logički radni prostor (*LogicalWorkspace).*

Izgled sučelja



Slika 1.8. *Organizacija glavnog prozora CPT aplikacije*

Područja koja su prikazana na slici su:

1. Izbornik (*Menu Bar*):

- ✓ Ovaj izbornik omogućava File,Edit, Options,View,Tools,Extensions, Help.
- ✓ Ovdje možete naći osnovne funkcije kao što su Open, Save, Print.
- ✓ Također imate pristup čarobnjaku (*Activity Wizard*) koji se nalazi u File izborniku.

2. Glavni alatni izbornik (*Main Tool Bar*):

- ✓ Ova alatna traka omogućuje prečace ikona na naredbe iz File izbornika, uključujući i čarobnjaka aktivnosti (*Activity Wizard*) i također (Copy, Paste, Undo, Redo, ZoomIn, ZoomReset, ZoomOut, crtanje paleta i Custom Device Dialog)
- ✓ Na desnoj strani, također možete pronaći gumb mrežne informacije (*Network Information*), koju možete koristiti za unos opisa trenutne mreže ili bilo kojeg teksta kojeg želite uključiti.

3. Zajednička alatna traka (*Common Tools Bar*): Ova traka omogućuje pristup najčešće korištenim alatima u radnom prostoru, a to su:

- ✓ Odaberi (*Select*)
- ✓ Premještanje rasporeda (*Move Layout*)
- ✓ Izbriši (*Delete*)
- ✓ Provjeri (*Inspect*)
- ✓ Dodaj jednostavan PDU (*Add Simple PDU*)
- ✓ Dodaj složeni PDU (*Add Complex PDU*)

4. Izbornik radnog prostora (*Workspace Type Bar*)

- ✓ Možete se prebacivati između fizičkog radnog prostora (*Physical Workspace*) i logičkog radnog prostora(*Logical Workspace*).

5. Radni prostor (*Workspace*):

- ✓ Ovo područje je mjesto gdje ćete stvoriti svoju mrežu, gledati simulacije te vidjeti mnoge vrste informacija i statističkih podataka.

6. Realni (*Realtime*) ili Simulacijski (*Simulation*) izbornik:

- ✓ Možete vršiti prijelaz između rada u stvarnom vremenu (*Realtime Mode*) i simulacijskog moda rada (*Simulation Mode*) sa kraticama iz ovog izbornika.

7. Izbor mrežnih komponenti (*Network Component Box*)

- ✓ Ovaj okvir omogućuje izbor uređaja (*device*) i veza (*connections*) za postavljanje na radni prostor.
- ✓ Također sadrži i okvir za izbor vrsta uređaja (*Device-Type Selection Box*) te (*DeviceSpecific Selection Box*)

8. Izbor vrsta uređaja (*Device Type Selection Box*):

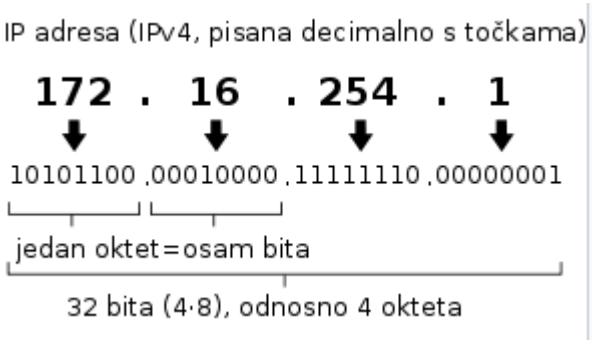
- ✓ Ovaj okvir sadržava vrste uređaja i priključaka na raspolaganju u Packet Traceru

- ✓ Određeni uređaji će se mijenjati ovisno o vrsti uređaja na kojeg ste kliknuli

9. Okvir za odabir uređaja (*Device Selection Box*):
   - ✓ Ovaj okvir je mjesto gdje možete izabrati koje točno uređaje želite staviti u svoju mrežu i koju vrstu veze želite upotrijebiti
10. Upravljač stvorenih programa (*Created Packet Window*)
   - ✓ Ovaj prozor upravlja paketima koje smo stavili u mrežu tijekom simulacije spoja.

2. Adresiranje u IPv4 računalnim mrežama

IP (Internet Protocol) adresa, u verziji 4 IP protokola (IPv4), je duga 32 bita. Lako je izračunati da je maksimalni broj različitih adresa 2^{32}, ili približno $4{,}3x10^9$ odnosno 4,3 milijardi adresa. Svako računalo koje je povezano na Internet mora imati jednoznačno dodijeljenu IP adresu. Te adrese su nužne da bi se paketi upućeni s izvorišnog računala mogli preusmjeriti do odredišnog. Taj postupak preusmjeravanja vrše uređaji specifične namjene koje nazivamo **usmjerivači** (*routers*). Kako je nužno da IP adrese budu jednoznačno dodijeljene, postoje međunarodne organizacije koje se brinu o raspodjeli IP adresnog prostora. Takva je organizacija ***The Internet Assigned Numbers Authority*** **(IANA)**. IANA zatim za određen raspon adresa zadužuje regionalne Internet registre, pa je za područje Europe zadužen RIPE Network Coordination Centre.



Slika 2.1. *Format Ipv4 adrese*

Dio bitova unutar IP adrese definira mrežu (*network*), a ostatak bitova definira dio adrese namjenjen označavanju mrežnih uređaja (*host*). Svaka IP adresa dolazi uz pripadajuću masku podmreže (*subnet mask*). Uz pomoć mrežne maske možemo razlučiti koji dio adrese predstavlja mrežni (network) dio, a koji dio označava adresu pojedinačnog uređaja unutar te mreže (host dio). Primjenjujući subnet masku dolazimo do adrese mreže i broadcast adrese. Primjenjujući subnet masku usmjerivači iz odredišne adrese IP paketa pronalaze adresu mreže, tj. saznaju gdje treba preusmjeriti paket kako bi stigao do odredišne mreže. Npr. Na datoj IP adresi mreže 192.168.1.0 sa mrežnom maskom 255.255.255.0 imamo jednu mrežu sa 256 (0-255) adresa. Adresu 0 ne možemo koristiti za adresiranje uređaja jer je to oznaka mreže, a ne možemo koristiti ni 255 jer se ta adresa koristi za *broadcasting* mreže, pa ukupno možemo iskoristiti 254 adrese za dodjelu uređajima na mreži.

192.168.1.0 = mrežna (*network*) adresa

192.168.1.1-254 = adrese za uređaje (*hosts*)

192.168.1.255 = *broadcast* adresa

Ova IP mreža može zapisati i kao **192.168.1.0/24**

Ovaj **/24** nam govori da prva 3 okteta označavaju mrežu, a zadnji oktet za označavanje uređaja.

255.255.255.0

11111111.11111111.11111111.00000000 3 x 8 = 24 i zato je to /24

Klase adresa

IP adrese su podjeljene u nekoliko klasa. Najčešće se koriste klase A, B i C. Svaka od klasa ima zadanu (*default*) *subnet* masku.

2 **Klasa A**(npr. **10.0.0.0/8**) ima 8 bitova rezerviranih za definiranje mrežnog dijela adrese. Zbog toga *default subnet* masku za klasu A označavamo kao **ip_adresa/8** Ostala 24 bita su rezervirana za označavanje uređaja (*hosts)*.
3 **Klasa B**(npr. **172.16.0.0/16**) ima 16 bitova za definiranje mrežnog dijela adrese. Možemo je pisati kao **ip_adresa/16** Ostalih 16 bitova označava uređaje.
4 **Klasa C** (npr. **192.168.1.0/24**) ima 24 bita rezervirana za definiranje mrežnog dijela adrese. Možemo je pisati kao **ip_adresa/24** Ostalih 8 bitova su rezervirana za označavanje uređaja.

Subnet maska se bilježi i u "***dotted decimal"*** notaciji. Ovakvo bilježenje IP adrese je nastalo radi lakšeg rada i upravljanja sa IP adresama.

Kao i drugi podaci u računalu, IP adresa i *subnet* maska su binarnog oblika. Kod *subnet* maske, bitovi mreže su označeni brojem 1, a bitovi koji označavaju uređaje su označeni nulom. Da bi ljudi lakše radili sa IP adresama i *subnet* maskama, njihova 32 bita su podjeljena u 4 grupe po 8 bitova koji su odvojene točkom. Svaka od tih grupa je iz binarnog prebačena u decimalni sustav.

Prema tome, zadana subnet maska za klasu A je iz oblika:

11111111000000000000000000000000

prešla u oblik

11111111.00000000.00000000.00000000

pa na kraju u oblik

255.0.0.0

Sukladno prethodnom, *default subnet* maska za klasu **B** je **255.255.0.0**,

a za klasu **C** je **255.255.255.0**.

Podmreža ili "*subnet"*

Podmreža ili ***subnet*** predstavlja manju mrežu unutar neke veće mreže.

Najmanja mreža, koje nema dodatnih podmreža, se naziva ***broadcast* domena**, što u osnovi predstavlja jednu lokalnu mrežu – **LAN**.

Unutar *broadcast* domene mrežni uređaji (računala, komunikacijska oprema,...) međusobno komuniciraju direktno, koristeći fizičke (***MAC- Media Access Control***) adrese.

Početna i završna adresa unutar podmreže imaju posebna značenja i ne koriste se kao adrese pojedinog mrežnog uređaja.

- Početna adresa je **adresa podmreže** (***Network ID***) koja identificira cijelu podmrežu. Kad želimo označiti cijelu podmrežu koristimo adresu podmreže.
- Završna ili ***broadcast*** adresa (***Broadcast ID***) je adresa na kojoj mrežni promet primaju sva računala unutar podmreže. Kad želimo poslati podatke svim uređajima u podmreži kao odredišnu adresu koristimo *broadcast* adresu.

Usmjerivači (***routers***) se koriste za povezivanje mreža. Njihova uloga je da promet primljen iz jedne mreže, a koji je namjenjen drugoj mreži, preusmjere prema toj drugoj mreži.

Broadcast promet je namjenjen svim računalima unutar samo jedne mreže. *Usmjerivači* ne preusmjeravaju promet s neke mreže na tu istu mrežu i ne prenose *broadcast* promet sa jedne mreže na drugu. Za usmjerivače uglavnom vrijedi pravilo da izoliraju *broadcast* domene.

U određenim slučajevima moguće je dopustiti usmjerivačima prijenos *broadcast* prometa (npr. za **DHCP** - ***Dynamic Host Configuration Protocol*** - automatsko dodjeljivanje mrežnih postavki mrežnim uređajima).

"Subnetiranje" se koristi za bolju kontrolu mrežnog prometa, omogućuje razvrstavanje mrežnog prometa na osnovu postavki mreže, te povećava sigurnost mreže tako što objedinjuje računala u logičke grupe.

Primjer 1. Određivanje mrežne adrese

Primjenom *subnet* maske na neku IP adresu uređaja razlučiti ćemo kojoj mreži taj uređaj (*host)* pripada, tj. koja je njegova mrežna adresa.

Zadana je IP adresa:

192.168.1.5

Ova adresa je privatna adresa klase C. Zadana (*default*) *subnet* maska za klasu C je /24.

/24 znači da su 24 bita rezervirana za definiranje mrežnog dijela IP adrese. Ostalih 8 (od ukupno 32) je rezervirano za *host* dio adrese.

Mrežni bitovi su predstavljeni sa 1, a *host* bitovi sa 0.

Ovo je binarni prikaz *default* maske podmreže klase C:

11111111111111111111111100000000

U *dotted decimal* zapisu je to:

255.255.255.0

Da bismo doznali tražene informacije potrebno je zadanu adresu također pretvoriti u binarni oblik:

192.168.1.5 = 11000000101010000000000100000101

Na dobivene binarne brojeve primjenjujemo logičku operaciju "**I"*(AND)***. Logička operacija "**I"** daje vrijednost 1 ako svi operandi imaju vrijednost 1.

Primjer logičke operacije ***AND***

- 0 AND 0 = 0
- 0 AND 1 = 0
- 1 AND 0 = 0
- 1 AND 1 = 1

Primjenjeno na naš primjer:

11000000101010000000000100000101 (192.168.1.5)
AND 11111111111111111111111100000000 (255.255.255.0)
11000000101010000000000100000000 (192.168.1.0)

Tražena adresa mreže je **192.168.1.0**

Privatne IP adrese

Unutar klasa IP adresa postoje određene adrese koje se ne usmjeravaju preko interneta. One su namjenjene samo za uporabu unutar izoliranih (privatnih mreža), a razlog njihova nastajanja je nedostatak jedinstvenih globalnih adresa u IPv4 adresnom prostoru. Takve adrese se nazivaju **privatne IP adrese**. Ovaj postupak omogućava mrežnim administratorima da unutar LAN-a mogu sasvim slobodno vršiti dodjelu adresa računalima iz skupa privatnih adresa bez obveze da se o tome konzultiraju sa bilo kime van organizacije.Ukoliko računalo sa privatnom adresom "želi" izaći na inteternet, njegova privatna adresa se zamjenjuje sa javnom adresom koju je moguće usmjeravati preko interneta. Taj postupak se naziva ***Network Address Translation (NAT).*** U ovom slučaju veliki broj računala unutar LAN mreže „izlazi" na Internet koristeći samo jednu (ili nekoliko) javnih IP adresa. Na taj se način danas rješava problem nedostatka javnih IPv4 adresa.

Slijedeći blokovi adresa su rezervirani za privatnu uporabu:

- Klasa A: 10.0.0.0/8 (od 10.0.0.0 do 10.255.255.255)
- Klasa B: 172.16.0.0/12 (172.16.0.0 to 172.31.255.255)
- Klasa C: 192.168.0.0/16 (192.168.0.0 to 192.168.255.255)

Kao što smo već spomenuli, unutar svake podmreže postoje adrese koje se ne mogu dodjeliti uređajima. Ako neka mrežna adresa u svom *host* dijeli ima sve 0, onda ona predstavlja **mrežnu adresu**, a ako u svom *host* dijelu ima sve 1, onda je ona ***broadcast*** adresa.

Primjer 2. Gubitak IP adresa

Trebamo adresirati 38 uređaja, adresom klase C.

Klasa C ima 8 bitova za označavanje uređaja (*host* bitovi), a prva 24 bita su mrežni bitovi. Ukupan broj hostova je $2^8 = 256$.

Označavanje počinje sa brojem 0, a ne sa brojem 1, pa to onda daje raspon od 0 do 255.

Baza je broj 2 zato što svaki bit može imati dvije vrijednosti: 0 ili 1. Eksponent je 8 jer imamo ukupno 8 bitova za određivanje broja uređaja (*host)*.

Ako uzmemo mrežnu adresu 192.168.1.0/24 i njome adresiramo 38 uređaja dobit ćemo adrese:

- 192.168.1.0 (ne koristi se jer je to adresa mreže)
- 192.168.1.1
- 192.168.1.2
- 192.168.1.3
- ...
- 192.168.1.38

Sve adrese od 192.168.1.39 do 192.168.1.254 su neiskorištene. (192.168.1.255 se ne koristi jer je to *broadcast* adresa). Na ovaj način smo izgubili 216 adresa.

Da bi izbjegli gubitak adresa iz prethodnog primjera, možemo zadanu IP mrežu dodatno *subnetirati* tj. jednu veliku mrežu podijeliti na više manjih mreža. To radimo na način da od *host* bitova uzmemo dio bitova i dodjelimo ih mrežnom dijelu adrese.

Primjer 3. *Subnetiranje*

Obraditi ćemo adrese iz prethodnog primjera: iz mreže 192.168.1.0 trebamo odrediti podmrežu (*subnet*) koja će imati dovoljno adresa za adresiranje 38 uređaja.

Mrežu 192.168.1.0 ćemo podjeliti u podmreže, tako što ćemo iz *host* dijela adrese uzeti odgovarajući broj bitova i dodijeliti ih mrežnom dijelu adrese.

Adresa 192.168.1.0 je adresa klase C, sa *default subnet* maskom /24.

Ostaje nam 8 bitova za uređaje.

Od tih 8, dio treba ostaviti za uređaje, a dio dodjeliti mreži.

Potrebno je adresiranje 38 uređaja.

- Ako od *host* dijela uzmemo 1 bit, moći ćemo adresirati 2 uređaja (2^1=2). To nam nije dovoljno.
- Ako uzmemo 2 bita, adresiramo 4 uređaja (2^2=4). Opet nedovoljno.
- Sa 3 bita adresiramo 8 uređaja (2^3=8). Opet nedovoljno.

- Sa 4 bita adresiramo 16 uređaja (2^4=16). Opet nedovoljno.
- Sa 5 bitova adresiramo 32 uređaja (2^5=32). Opet nedovoljno.
- Sa 6 bitova adresiramo 64 uređaja (2^6=64). Sa 6 bitova možemo adresirati 64 uređaja, što je više nego dovoljno za adresiranje potrebnih 38 uređaja.

Rješenje je da od *host* bitova, 6 bita ostavimo za adresiranje uređaja, a 2 bita možemo pripojiti mrežnom dijelu. Sa 2 bita **(n=2)** dodjeljena mreži 192.168.1.0, može se dobiti ukupno 4 podmreže.

Ukupan broj podmreža je 2^n**=4**

Do iskoristivog dijela za adresiranje uređaja (*hosts*) dođemo po formuli:

$$2^{\text{broj preostalih nula u subnet maski}} - 2$$

(jer 2 adrese otpadaju na mrežnu i *broadcast* adresu)

4 podmreže koje smo dobili imaju svaka po 64 adrese, a mogu se adresirati 62 uređaja i to su:

- 1. 192.168.1.0 – 192.168.1.63
- 2. 192.168.1.64 – 192.168.1.127
- 3. 192.168.1.128 – 192.168.1.191
- 4. 192.168.1.192 – 192.168.1.255

U prethodnom primjeru smo vidjeli da je ista *subnet* maska primjenjena na sve podmreže. Ovakav način podjele velike mreže u više manjih mreža jednake veličine naziva se **FLSM (*Fixed Length Subnet Mask*)**. Problem može nastati ako je jednu veliku mrežu potrebno podijeliti u više manjih mreža, ali sa različitim brojem uređaja u svakoj od njih.

Za riješenje tog problema se koristi **VLSM** (***Variable Length Subnet Mask***). VLSM predstavlja način podjele IP adresa prema pojedinim zahtjevima svake mreže, a ne prema nekim općenitim normama adresiranja.

Primjer 4. VLSM

Recimo da se pored potrebe adresiranja 38 uređaja i jednoj mreži, što smo obradili u prethodnom primjeru, javila potreba adresiranja još nekoliko podmreža sa različitim brojem uređaja:

- a) 38 uređaja (već obrađeno)
- b) 15 uređaja
- c) 10 uređaja
- d) 2 uređaja.

Izračun za mrežu b) sa 15 uređaja

Prva iskoristiva mreža iz prijašnjeg primjera iskorištena je za adresiranje 38 uređaja.

Nama je potrebno adresirati još 3 mreže sa po 15, 10 i 2 uređaja. Prvo ćemo "subnetirati" za mrežu sa najvećim brojem uređaja (15) i dalje padajući po veličini.

Za adresiranje 15 uređaja potrebno je od *host* dijela adrese uzeti 5 bita za adrese uređaja, a 3 (n=3) nam ostaju za adresiranje same podmreže. Sa 5 bita dobijemo ukupno $2^5=32$ adrese tj. možemo adresirati 30 uređaja, (nama je potrebno 15). Nismo uzeli 4 bita jer prema formuli $2^n-2=2^4-2=14$ ostaje nam samo 14 *host* adresa, što nije dovoljno.

Obzirom da smo za prvu podmrežu iskoristili adrese 192.168.1.0 – 192.168.1.63 , prva sljedeća adresa mreže koju možemo koristiti je 192.168.1.64.

Subnet maska nove podmreže je /27 (24 *default* bita + 3 bita podmreže koja su ostala nakon što smo 5 bita rezervirali za uređaje) ili 255.255.255.224

Raspon adresa sljedeće iskoristive podmreže je: 192.168.1.64 – 192.168.1.95

Sumarno:

- 192.168.1.64/27 (adresa mreže i ne koristi se za adresiranje uređaja)
- 192.168.1.65/27 (prva iskoristiva adresa za uređaje)
- ...
- 192.168.1.94/27 (zadnja iskoristiva adresa za uređaje)
- 192.168.1.95/27 (*broadcast* adresa mreže i ne koristi se za uređaje)

Izračun za mrežu c) sa 10 uređaja

Trebamo adresirati 10 uređaja.

Zadnja iskorištena adresa je 192.168.1.95

Za adresiranje 10 uređaja trebamo 4 bita ($2^n-2=2^4-2=14$ mogućih *host* adresa). Znači n=4

Subnet maska je /28 (24 *default* bita + 4 bita podmreže koja su ostala nakon što smo 4 bita rezervirali za uređaja).

Sumarno:

- 192.168.1.96/28 (adresa mreže i ne koristi se za uređaje)
- 192.168.1.97/28 (prva iskoristiva *host* adresa)
- ...
- 192.168.1.110/28 (zadnja iskoristiva *host* adresa)
- 192.168.1.111/28 (*broadcast* adresa mreže i ne koristi se za uređaje)

Izračun za mrežu d) sa 2 uređaja

2 bita za uređaje = 2 *host* adrese (2^n-2=2^2-2=2)

6 bita za mrežu tj./30 *subnet* maska (24 + 6)

- 192.168.1.112/30 (adresa mreže)
- 192.168.1.113/30 (1. *host* adresa)
- 192.168.1.114/30 (2. *host* adresa)
- 192.168.1.115/30 *(broadcast* adresa)

Korištenjem VLSM (*Classless Interdomain Routing*- CIDR) mehanizma ukupno je utrošeno 116 (od 256 adresa), tj. sačuvano je 140 adresa za dalje korištenje.

U ovim primjerima su korištene IP adrese klase C koje se i inače najčešće koriste u privatnim mrežama. Na isti način se vrši podmrežavanje (*subnetting*) za bilo koju klasu adresa.

Classfull adresiranje koristi *default subnet* masku za neku adresu. Nedostatak mu je što se, bez obzira na stvarne potrebe, troši prevelik broj IP adresa. Ako se ukine sustav razvrstavanja adresa po klasama, moguće je uštediti te adrese.

Classless Interdomain Routing(CIDR) je uveden kao mehanizam koji poboljšava iskoristivost adresnog prostora i skalabilnost usmjeravanja prometa preko interneta. Kod **CIDR**-a je napravljen odmak od tradicionalne podjele mreža na klase, te su mreže predstavljene IP adresom i brojem bitova u *subnet* maski (npr. 192.168.1.0/24). [4]

Zadaci:

Pitanje 1: Da li je 192.168.1.153 /27 adresa uređaja (*host* adresa)?

Pitanje 2: Izaberite ispravnu mrežnu masku kako bi napravili 4 mreže sa zadanim brojem korisnika

- 44 korisnika
- 60 korisnika
- 22 korisnika
- 12 korisnika

a) 255.255.255.128
b) 255.255.255.192
c) 255.255.255.224
d) 255.255.255.240
e) 255.255.255.248

Pitanje 3. Koja od slijedećih je *host* adresa?

a) 192.168.2.224 /28
b) 192.168.2.47 /28
c) 192.168.2.160 /28
d) 192.168.2.192 /28

Rješenje:

Pitanje 4. Kojoj mreži pripada IP adresa 200.168.6.101 sa *subnet* maskom 255.255.255.224 ?

Rješenja:

Pitanje 1: Da li je 192.168.1.153 /27 regularna *host* adresa

Da bi to riješili moramo raspisati mreže

/27 znači da je maska podmreže 11111111.11111111.11111111.11100000 tj. 255.255.255.224

Inkrement (ukupan broj adresa adresa po mreži) je 32, a imamo ukupno 8 mreža.

Znači mreže idu:

192.168.1.0-31

192.168.1.32-63

192.168.1.64-96

192.168.1.96-127

192.168.1.128-159 ----- 192.168.1.153 JE KORISNA *HOST* ADRESA

192.168.1.160-191

192.168.1.192-223

192.168.1.224-255

Pitanje 2: Izaberite ispravnu mrežnu masku kako bi napravili 4 mreže sa zadanim brojem korisnika

Prvo raspišemo to binarno i nađemo inkrement

a) 255.255.255.128 10000000 MN=128

b) 255.255.255.192 11000000 **MN=64** - broj korisnih hostova 62 u 4 mreže

c) 255.255.255.224 11100000 MN=32

d) 255.255.255.240 11110000 MN=16

e) 255.255.255.248 11111000 MN=8

Pitanje 3. Koja od slijedećih je korisna *host* adresa?

e) 192.168.2.224 /28 Ne može jer je to network adresa
f) 192.168.2.47 /28 Ne može jer je to broadcast adresa
g) 192.168.2.160 /28 Ne može jer je to network adresa
h) 192.168.2.192 /28 Ne može jer je to network adresa

Rješenje:

NITI JEDNA!!!!

ukradena 4 bita, Inkrement (ukupan broj adresa) =16, mreža imamo 2^4 = 16. Mreže su:

192.168.2.0-15	192.168.2.64-95	192.168.2.128-143	192.168.2.192-207
192.168.2.16-31	192.168.2.80-95	192.168.2.144-159	192.168.2.208-223

192.168.2.32-47	192.168.2.96-111	192.168.2.160-175	192.168.2.224-239
192.168.2.48-63	192.168.2.112-127	192.168.2.176-191	192.168.2.240-256

Pitanje 4. Kojoj mreži pripada IP adresa 200.168.6.101 sa *subnet* maskom 255.255.255.224 ?

Uvijek prvo tražimo inkrement iz *subnet* maske. Ovdje gledamo samo .224 tj. 11100000 Inkrement je 32 imamo, a imamo ukupno 8 mreža jer imamo 3 posuđene jedinice, a 2^3=8. Mreže su:

200.168.6.0, 200.168.6.32, 200.168.6.64, **200.168.6.96**, 200.168.6.128, 200.168.6.160, 200.168.6.192, 200.168.6.224

ADRESA PRIPADA 200.168.6.96 MREŽI

3. Statičko usmjeravanje

Usmjerivač ili *router* je uređaj koji usmjerava mrežni promet između **različitih mreža**. Svako njegovo sučelje (*interface*) mora imati svoju IP adresu koja pripada IP mreži spojenoj na to sučelje. Generalno je pravilo da se prva slobodna *host* adresa mreže dodjeljuje sučelju usmjerivača. Tu adresu na sučelje usmjerivača postavlja mrežni administrator.

Svakom računalu u mreži potrebno je obznaniti adresu tzv. „*gateway*" usmjerivača (*router*) prema kojemu će računalo slati pakete ukoliko treba komunicirati sa nekim uređajem van njegove mreže. Računalu je poznata IP adresa „udaljenog" uređaja, te adresa *gateway* usmjerivača kojemu šalje podatke. *Gateway* usmjerivač je taj koji iz odredišne IP adrese paketa (IP adresa „udaljenog" uređaja) pronalazi put prema toj mreži i šalje paket prema njoj. Usmjerivači ne usmjeravju IP pakete na razini IP adresa uređaja, već mreža.

Što to znači?

Iz odredišne IP adrese paketa i pripadajuće mrežne maske, usmjerivač pronalazi odredišnu **mrežnu adresu** te šalje IP paket prema toj mreži! Svaki usmjerivač zna točno koje mreže se nalaze direktno spojene na njegova sučelja, a za ostale (udaljenije) mreže stalno radi i obnavlja tzv. **tablice usmjeravanja** (*routing tables*). Što je to?

Usmjerivači osim što stalno prespajaju promet unutar mreža koje direktno povezuju, također stalno međusobno komuniciraju sa ostalim usmjerivačima na mreži (internetu) pomoću **protokola usmjeravanja** kako bi znali gdje proslijediti pakete za bilo koju mrežu na svijetu i temeljem tih informacija grade svoju tablicu usmjeravanja. Obzirom da se do neke mreže može doći na nekoliko načina, oni stalno traže "najbolju" rutu. Nekad je „najbolja" bila ona koja ide preko najmanje skokova (*hop*), a danas se u obzir uzimaju i drugi faktori poput širine pojasa, kašnjenja ili bilo kojeg drugog parametra prijenosa. Protokoli usmjeravanja također stalno nadgledaju da li je određena ruta "*zdrava"* tj. da nije u prekidu. Ukoliko usmjerivač treba proslijediti informaciju za neku mrežu o kojoj nema informacija, on odbacuje paket.

Ima više različitih protokola usmjeravanja, a generalno ih dijelimo na IGP (*Interior Gateway Protocols*) i EGP (*Exterior Gateway Protocols*). Koja je razlika?

Tablice usmjeravanja ograničene su svojom veličinom. Na svijetu postoji previše mreža da bi sve stale u tablicu usmjeravanja svakog usmjerivača. Zbog toga imamo skalabilnost cijelog sustava Interneta. Skalabilnost počinje dodjelama javnih IP adresa, pa su tako određeni blokovi IP adresa za Ameriku, Europu, ..itd. Unutar Europe, određeni blok adresa dodjeljuje se svakoj pojedinoj regiji. Unutar regije, regionalni upravitelj dodjeljuje blokove adresa Autonomnim Sustavima (AS) koji su uglavnom javni pružatelji pristupa Internetu ali i velike kompanije, organizacije ili javne ustanove koje to zatraže. IGP protokoli koriste se za razmjenu usmjerivačkih informacija unutar jednog autonomnog sustava mreža (npr. Unutar sveučilišta ili unutar neke organizacije ili čak javnog davatela usluga pristupa Internetu), dok EGP protokoli služe za međusobno povezivanje tih Autonomnih Sustava. Internet, koji je najveća globalna mreža, sastoji se od međusobno povezanih Autonomnih Sustava. Najpoznatiji IGP protokoli su RIP, OSPF i IS-IS protokoli, dok je kod EGP protokola najpoznatiji BGP.

Za vrlo male mreže koje nisu spojene na Internet ne moramo uopće koristiti usmjerivačke protokole već sve rute možemo ručno upisati u usmjerivač i to nazivamo statički usmjeravanjem.

Statičko usmjeravanje (*static routing*) – svaku rutu upisujemo ručno u usmjerivačku tablicu usmjerivača. Format upisivanja je slijedeći:

ip adresa mreže, njena mrežna maska, ip adresa sučelja na koji paket treba proslijediti.

Prednosti statičkog usmjeravanja

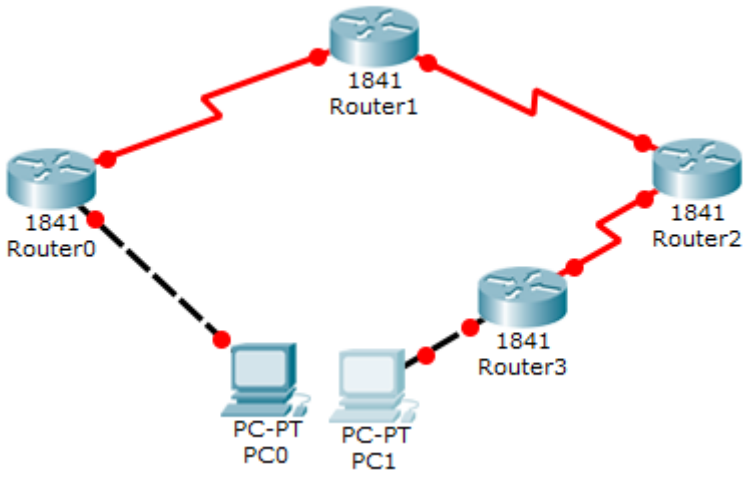
- Jednostavno
- najsigurniji način jer usmjerivači međusobno ne izmjenjuju informacije
- Ne zahtjeva veliki rad CPU-a u usmjerivaču ili veliku memoriju.

Loše strane

- Dobro je samo za male mreže
- Ako neka veza padne, promet se neće automatski prebaciti na neki drugi pravac (ako nije kreirana i statička *backup* ruta)

Zadatak - Statičko usmjeravanje

Postavi IP adrese svih sučelja, te statičko usmjeravanje na svim usmjerivačima u mreži kako bi se omogućila komunikacija između PC0 i PC1



Slika 3.1. *Zadana mrežna topologija*

Uređaj	Početno sučelje	Odredišno sučelje	IP Adresa
PC0	FastEthernet0	Router0 FastEthernet0/0	10.0.0.2/8
Router0	FastEthernet0/0	PC0 FastEthernet0	10.0.0.1/8
Router0	Serial 0/0/0	Router1 serial0/0/0	192.168.0.253/30
Router1	Serial0/0/0	Router0 Serial0/0/0	192.168.0.254/30
Router1	Serial0/0/1	Router2 Serial0/0/0	192.168.0.249/30
Router2	Serial0/0/0	Router1 Serial0/0/1	192.168.0.250/30
Router2	Serial0/0/1	Router3 Serial0/0/0	192.168.0.245/30
Router3	Serial0/0/0	Router2 Serial0/0/1	192.168.0.246/30
Router3	FastEthernet0/0	PC1 FastEthernet0	20.0.0.1/8
PC1	FastEthernet0	Router1 FastEthernet0/0	20.0.0.2/8

Tablica 3.1. Adresiranje sučelja u zadatku

Dodjeli IP adresu računalima i usmjerivačima

```
Router#configure terminal
```
Naredba za ulazak u globalni konfiguracijski mod.

```
Router(config)#interface serial 0/0/0
```
Naredba za ulazak u konfiguraciju sučelja (*interface*).

```
Router(config-if)#ip address 192.168.0.253 255.255.255.252
```
Naredba za dodjeljivanje IP adrese sučelju. Za serijsko sučelje obično koristimo IP adrese iz /30 podmreže.

```
Router(config-if)#clock rate 64000
Router(config-if)#bandwidth 64
```

Zadnja dva parametra (*clockrate* i *bandwidth*) u relnosti kontroliraju protok informacija po serijskoj vezi i najčešće se postavljaju od strane pružatelja usluga. Obzirom da se ovdje radi o laboratorijskoj simulaciji, ne trebamo brinuti o tome već možemo koristiti ove vrijednosti..

```
Router(config-if)#no shutdown
```
Naredba za „podizanje“ sučelja u aktivno stanje.

```
Router(config-if)#exit
```
Naredba za povratak u globalni konfiguracijski mod.

Kada smo podigli sva sučelja i dodjelili im adrese, tek sad postavljamo statičko usmjeravanje. Postoje dvije oblika naredbe za konfiguraciju statičkog rutiranja:

```
Router(config)# ip route destination_network_# [subnet_mask] IP_address_of_next_hop_neighbor [administrative_distance] [permanent]
```

ili

```
Router(config)# ip route destination_network_# [subnet_mask] interface_to_exit [administrative_distance] [permanent]
```

U zagradama [ ] su navedeni opcijski parametri, tj. nisu obavezni već se koriste ako treba

ip route - Ovo je osnovna naredba koja dodaje novu rutu u tablicu usmjeravanja.

destination_network_#[subnet_mask] - Ovo je prvi parametar. Specificira *destination network address*, tj. adresu koju ćemo prosljeđivati. Ukoliko koristimo "subnetiranu" mrežu moramo dodati i masku podmreže. "Subnetirane" mreže su manje mreže napravljene od jedne veće mreže.

IP_address_of_next_hop_neighbor / interface_to_exit - Ovaj parametar pokazuje put gdje trebamo poslati informaciju. Obje naredbe koriste poseban način za dodjelu ove vrijednosti. Prva naredba određuje IP adresu slijedećeg skoka. Ona kaže usmjerivaču da ako zaprimi paket za mrežu koju smo postavili u predhodnom koraku, proslijedi taj paket na IP adresu slijedećeg skoka.

Druga naredba umjesto IP adrese slijedećeg skoka specificira sučelje. (u biti ista stvar)

administrative_distance - Administrativna udaljenost označava pouzdanost veze. Ruta sa manjom udaljenošću je u prednosti pri odabiru rute. Po standardnim postavkama, ako se koristi IP adresa od *next hop neighbor* , AD vrijednost je 1, a ako smo koristili izlazno sučelje, AD vrijednost je 0. Ovaj parametar omogućava nam da i kod statičkog usmjeravanja kreiramo višestruke rute (*multiple static routes)* za istu destinaciju. Da bi kreirali pričuvnu rutu (*backup path)*, moramo toj ruti dodjeliti AD vrijednost veću od primarne radne (*default route*), npr. 2 ili 3. Sa ovakvom konfiguracijum usmjerivač će koristiti primarni put dokle god on funkcionira.

Permanent - Kada neka ruta ispadne iz rada, usmjerivač je miče iz tablice. „Permanent" parametar će zadržati ovu rutu u tablici čak ako ona i „padne". To je opcijski parametar (ne moramo ga uopće postavljati ako ne želimo) koji možemo koristiti iz sigurnosnih razloga kada nikad ne želimo da paketi idu drugom rutom.

Sad kad znamo naredbe, idemo postaviti statičko usmjeravnje na našu mrežu.

Ne moramo konfigurirati direktno spojene mreže! Npr. Router 0 je direktno spojen sa PC0, a ruter3 je spojen sa PC1

Router0

```
Router(config)#ip route 20.0.0.0 255.0.0.0 192.168.0.254
```

Ova naredba kaže usmjerivaču da ako primi paket za 20.0.0.0 mrežu, da ga proslijedi na 192.168.0.254. Mreža 10.0.0.0 je direktno spojena na ovaj usmjerivač pa ne moramo konfigurirati dolaznu vezu za nju.

Router1

```
Router(config)#ip  route 10.0.0.0 255.0.0.0 192.168.0.253
Router(config)#ip  route 20.0.0.0 255.0.0.0 192.168.0.250
```

Sa ovog usmjerivača do obje mreže se može doći samo preko drugih usmjerivača, pa moramo definirati oba pravca (za obje mreže 10.0.0.0 i 20.0.0.0.)

Router2

```
Router(config)#ip  route 10.0.0.0 255.0.0.0 192.168.0.249
Router(config)#ip  route 20.0.0.0 255.0.0.0 192.168.0.246
```

Jednako kako i za usmjerivač Router1, i ovdje moramo kreirati oba pravca

Router3

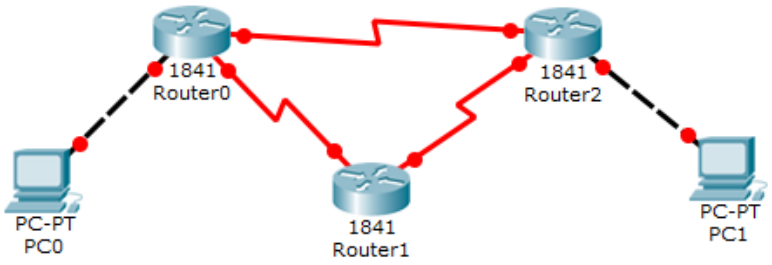
```
Router(config)#ip  route 10.0.0.0 255.0.0.0 192.168.0.245
```

Mreža 20.0.0.0 je direktno spojena pa samo trebamo konfigurirati mrežu 10.0.0.0 na ovom usmjerivaču.

4. RIP (*Routing Information Protocol*) potokol usmjeravanja

RIP protokol - svaki usmjerivač pomoću periodičnog broadcasta (svakih 30 sekundi) šalje ostalim usmjerivačima popis svih mreža koje su na njega direktno spojene. Ove *broadcast* poruke zovu se „*routing updates*“. Kada usmjerivač sazna za neku novu mrežu, on je odmah uvrsti u svoju tablicu usmjeravanja (*routing table*). Usmjerivač provodi metriku i gleda koliko skokova (*hops*) mu treba do neke mreže sa novom rutom. Ukoliko ima manje skokova, izbacuje staru, a uvrštava novu rutu u tablicu usmjeravanja. Ukoliko je nova ruta lošija, ignorira je. Ukoliko je vrijednost jednaka, usmjerivač resetira *timer* za tu rutu. Prilikom slanja periodičkih poruka osvježavanja stanja (*update*) usmjerivač ne traži povratnu obavjest (potvrdu) od drugih usmjerivača da su primili obavijest. RIP protokol se oglašava sa svih sučelja (*interface*) usmjerivača. Ipak, RIP protokol nam daje mogućnost da neko sučelje proglasimo „pasivnim“ i onda ono ne šalje osvježavajuće poruke stanja (*routing update*). **„Split horizon“** – funkcija koja omogućava da usmjerivač ne oglašava rutu nazad prema usmjerivaču koji mu je istu prethodno poslao, kako bi se izbjegle petlje.

Zadatak - Konfiguracija RIP protokola usmjeravanja



Slika 4.1. *Zadana mrežna topologija*

Uređaj	Sučelje	IP Adresa	Spojen sa
PC0	Fast Ethernet	10.0.0.2/8	Router0 Fa0/1
Router0	Fa0/1	10.0.0.1/8	PC0 Fast Ethernet
Router0	S0/0/1	192.168.1.254/30	Router2 S0/0/1
Router0	S0/0/0	192.168.1.249/30	Router1 S0/0/0
Router1	S0/0/0	192.168.1.250/30	Router0 S0/0/0
Router1	S0/0/1	192.168.1.246/30	Router2 S0/0/0
Router2	S0/0/0	192.168.1.245/30	Router1 S0/0/1
Router2	S0/0/1	192.168.1.253/30	Router0 S0/0/1
Router2	Fa0/1	20.0.0.1/30	PC1 Fast Ethernet
PC1	Fast Ethernet	20.0.0.2/30	Router2 Fa0/1

Tablica 4.1. Adresiranje sučelja u zadatku

Korak 1 - Dodjeljivanje IP adresa i *default gateway* adrese računalima (PC)

Dvostruki "klik" na **PC0**, pa jednostruki "klik" na **Desktop** menu, pa "klik" na **IP Configuration**. Dodjeli IP adresu 10.0.0.2/8 za PC0, te dodijeli adresu 10.0.0.1/8 za *default gateway*. Ponovi proces za PC1 i dodjeli mu IP adresu 20.0.0.2/8 , te adresu 20.0.0.2/8 za njegov *default gateway*.

Korak 2 – Dodjeljivanje IP adresa sučeljima usmjerivača

Dvostruki "klik" na **Router0**, pa "klik" na **CLI,** pa pritisni **Enter key** kako bi pristupio u „*command prompt*" od **Router0**.

Tri sučelja (*FastEthernet0/0*, *Serial0/0/0* i *Serial0/0/1)* od usmjerivača **Router0** se koriste u ovoj topologiji. Po tvorničkim postavkama (*default*) sučelja na usmjerivačima ostaju administrativno isključena (*administratively down*) za vrijeme uključivanja uređaja. Prvo moramo konfigurirati IP adrese svakog sučelja, prije nego što ih možemo početi koristiti za usmjeravanje prometa. Postavljanje IP adrese i ostalih parametara sučelja vrši se iz tzv. „***Interface*" moda**. „*Interface*" modu se pristupa iz **globalnog konfiguracijskog moda**.

Sljedeće naredbe se koriste da bi se pristupilo globalnom konfiguracijskom modu.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

Iz globalnog konfiguracijskog moda prelazimo u *Interface* mod, a u njemu vršimo konfiguraciju sučelja (*interface*). Slijedećim naredbama izvršiti ćemo dodavanje IP adrese na sučelje FastEthernet0/0.

```
Router(config)#interface fastEthernet 0/0  naredbom ulazimo u „interface" mod
Router(config-if)#ip address 10.0.0.1 255.0.0.0  naredba dodjeljuje IP adresu sučelju
Router(config-if)#no shutdown  naredba postavlja („podiže") sučelje u radno stanje
Router(config-if)#exit  naredba za povratak u globalni konfiguracijski mod
Router(config)#
```

Serijsko sučelje treba za konfiguraciju još dva dodatna parametra ***clock rate*** i ***bandwidth***. Svaki serijski kabel ima dva različita kraja DTE i DCE. Ovi parametri se uvijek konfiguriraju samo na DCE kraju.

Možemo koristiti naredbu ***show controllers interface*** iz „*privileged*" moda da bi provjerili koji kraj kabela je spojen na naše sučelje.

```
Router#show controllers serial 0/0/0
Interface Serial0/0/0
Hardware is PowerQUICC MPC860
DCE V.35, clock rate 2000000
[Output omitted]
```

Četvrta linija odgovora koje dobijemo potvrđuje da je na sučelje spojen DCE završetak kabela.

Sada imamo sve potrebne informacije i možemo krenuti u dodavanje IP adrese serijskom sučelju.

```
Router#configure terminal  Naredba za ulaz u globalni konfiguracijski mod
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0/0/0  Naredba za ulaz u interface mod
Router(config-if)#ip address 192.168.1.249 255.255.255.252  Naredba kojom se dodaje
IP adresa sučelju
Router(config-if)#clock rate 64000  Naredbe kojima se postavljaju dodatni parametri
Router(config-if)#bandwidth 64  Naredbe kojima se postavljaju dodatni parametri
Router(config-if)#no shutdown  Naredba kojom su sučelje „podiže“ u radno stanje
Router(config-if)#exit  Naredba kojom se vraćamo u globalni konfiguracijski mod
Router(config)#interface serial 0/0/1
Router(config-if)#ip address 192.168.1.254 255.255.255.252
Router(config-if)#clock rate 64000
Router(config-if)#bandwidth 64
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
```

Iste naredbe koristimo za dodjelu IP adresa i na ostalim usmjerivačima. Parametre ***clock rate*** i ***bandwidth*** treba definirati samo na DCE kraju kabela kod serijskih sučelja. Slijedećim naredbama postavljaju se IP adrese sučelja na usmjerivačima Router1 i Router2.

Router1

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 192.168.1.250 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/1
Router(config-if)#ip address 192.168.1.246 255.255.255.252
Router(config-if)#clock rate 64000
Router(config-if)#bandwidth 64
Router(config-if)#no shutdown
Router(config-if)#exit
```

Router2

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fastEthernet 0/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 192.168.1.245 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/1
Router(config-if)#ip address 192.168.1.253 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
```

Sada usmjerivači znaju koje su mreže direktno spojene na njihova sučelja.

Usmjerivači neće sami od sebe početi razmjenjivati te informacije. Mi moramo pokrenuti RIP protokol usmjeravanja koji će pokrenuti taj proces.

Korak 3 - Konfiguracija RIP protokola

Konfiguracija RIP protokola vrlo je jednostavna. Zahtjeva samo dva koraka.

- Omogućavanje RIP protokola usmjeravanja iz globalnog konfiguracijskog moda.
- Obznaniti RIP protokolu koje mreže želimo oglašavati.

Idemo ga sada pokrenuti na Router0

Router0

```
Router0(config)#router rip
Router0(config-router)# network 10.0.0.0
Router0(config-router)# network 192.168.1.252
Router0(config-router)# network 192.168.1.248
```

router rip - naredba koja kaže usmjerivaču da pokrene RIP protokol usmjeravanja.

network - naredba koja nam omogućava da specificiramo mreže koje želimo oglašavati. Trebamo specificirati samo mreže koje su direktno spojene na usmjerivač.

To je sve što nam treba za pokretanje RIP protokola usmjeravanja. Idemo sada ponoviti radnju i za sve ostale usmjerivače u mreži.

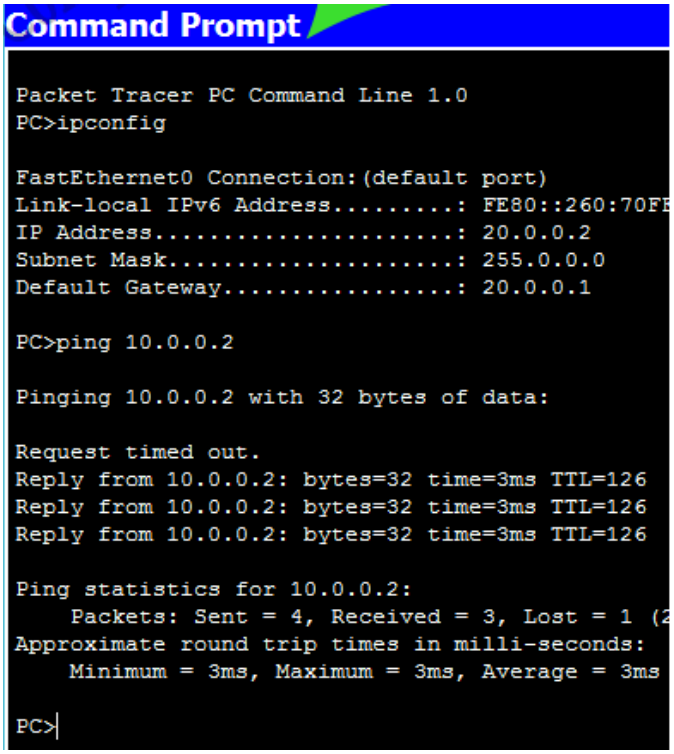
Router1

```
Router1(config)#router rip
Router1(config-router)# network 192.168.1.244
Router1(config-router)# network 192.168.1.248
```

Router2

```
Router2(config)#router rip
Router2(config-router)# network 20.0.0.0
Router2(config-router)# network 192.168.1.252
Router2(config-router)# network 192.168.1.244
```

I to je to! Sada nam samo preostaje da provjerimo funkcionalnost na način da naredbom ***ping*** iz *command prompt* sučelja testiramo vezu između **PC1** i **PC0**.



Slika 4.2. *Izgled Command Prompt sučelja nakon uspješne komunikacije*

RIP protokol automatski upravlja rutama. Ukoliko jedna ruta ispadne iz rada, on će promet automatski prebaciti na drugu rutu. Da bi objasnili ovaj proces dodali smo još jednu rutu u našu mrežu.

Trenutačno postoje dvije rute (staze) između PC0 i PC1.

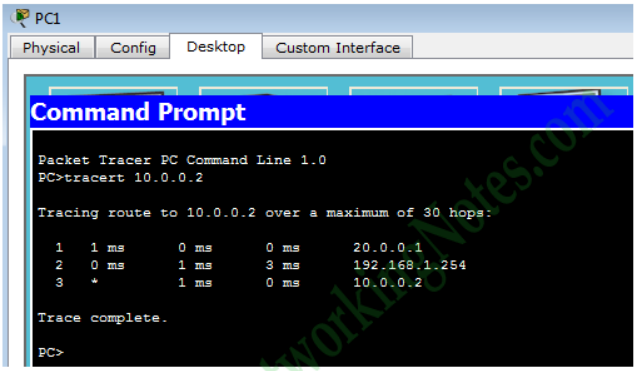
Ruta 1

PC0 [10.0.0.2] <==> Router0 [FastEthernet0/1 – 10.0.0.1] <==> Router0 [Serial0/0/1 – 192.168.1.254] <==> Router2 [Serial 0/0/1 – 192.168.1.253] <==> Router2 [FastEthernet0/0 – 20.0.0.1] <==> PC1 [20.0.0.2]

Ruta 2

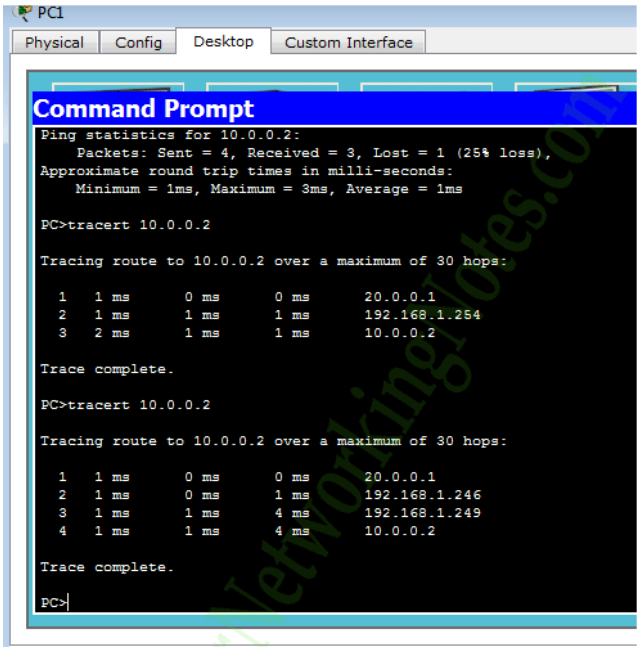
PC0 [10.0.0.2] <==> Router0 [FastEthernet0/1 – 10.0.0.1] <==> Router0 [Serial0/0/0 – 192.168.1.249] <==> Router1 [Serial 0/0/0 – 192.168.1.250] <==> Router1 [Serial 0/0/1 – 192.168.1.246] <==> Router2 [Serial 0/0/0 – 192.168.1.245] <==> Router2 [FastEthernet0/0 – 20.0.0.1] <==> PC1 [20.0.0.2]

Po pravilu (*by default*) RIP će koristiti rutu koja ima manje skokova između izvora i odredišta. U našem slučaju to je Ruta1. Pomoću naredbe ***tracert*** možemo to provjeriti.



Slika 4.3. *Uporaba naredbe „tracert“*

Sada pretpostavimo da nam je Ruta1 iz nekog razloga ispala iz rada. To možemo simulirati brisanjem kabela spojenog između **Router0 [s0/0/1]** i **Router2 [s0/0/1]**. Što će se dogoditi? Obzirom da smo pokrenuli RIP protokol, on će automatski preusmjeriti promet. To možemo provjeriti ponovnom uporabom naredbe ***tracert*** .



Slika 4.4. *RIP preusmjerenje na drugu rutu*

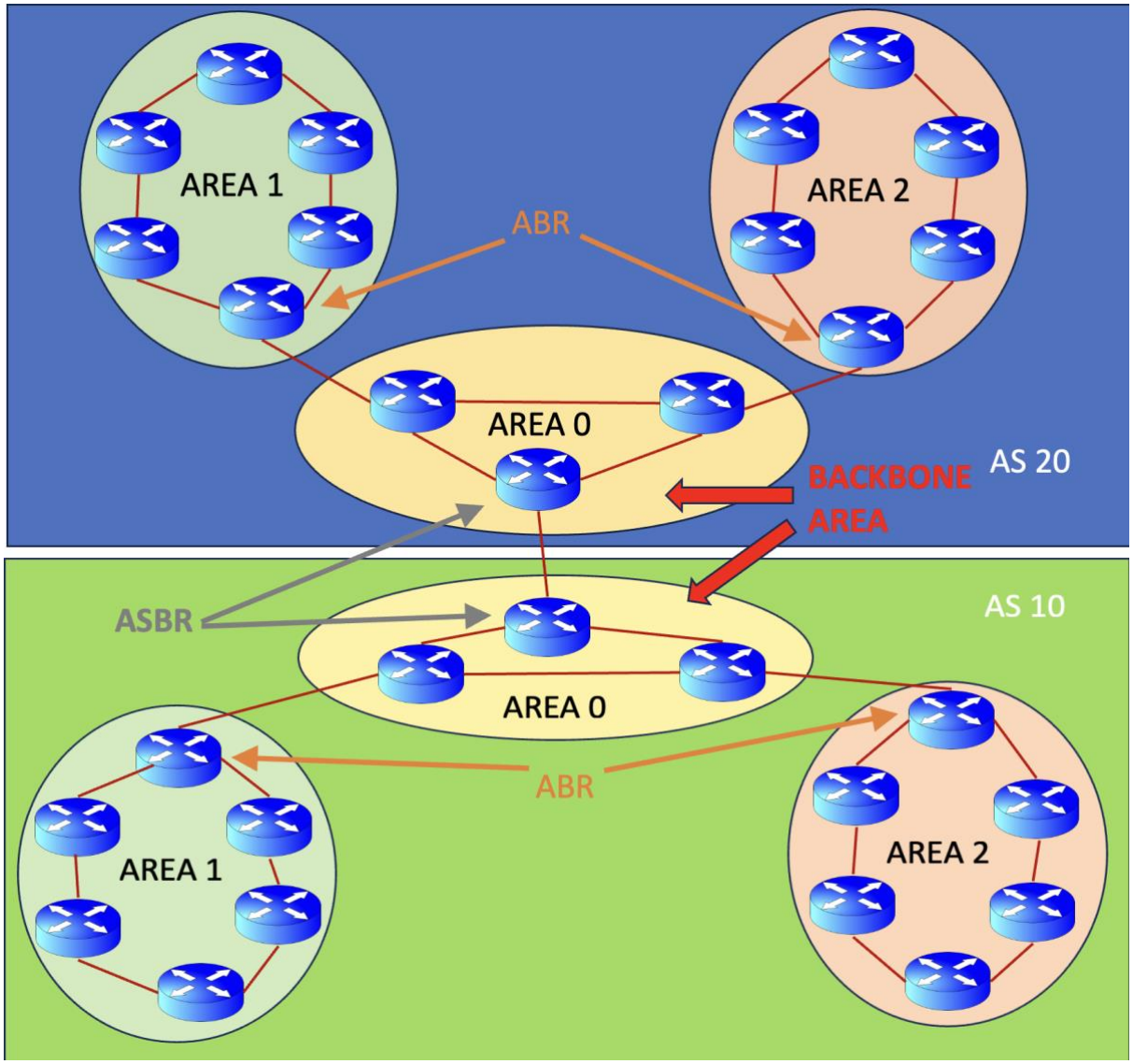
Sumirane naredbe za konfiguraciju RIP protokola usmjeravanja

Naredba	**Opis**
Router(config)#router rip	Pokreni RIP protokol usmjeravanja
Router(config-router)#network a.b.c.d	Dodaj a.b.c.d mrežu u RIP oglašavanje ruta
Router(config-router)#no network a.b.c.d	Makni a.b.c.d mrežu iz RIP oglašavanja
Router(config-router)#version 1	Pokreni RIP protokol verzije jedan (default)
Router(config-router)#version 2	Pokreni RIP protokol verzije dva
Router(config-router)#no auto-summary	Po default-u RIPv2 automatski sumira mreže u njihove klasne granice. Ova naredba će to isključiti.
Router(config-router)#passive-interface s0/0/0	RIP neće odašiljati *routing update* poruke sa ovog sučelja
Router(config-router)#no ip split-horizon	Onemogućuje *split- horizon* funkciju (Default - uključeno stanje bez da koristimo naredbe)
Router(config-router)#ip split-horizon	Ponovno omogućuje spilt- horizon funkciju
Router(config-router)#timers basic 30 90 180 270 360	Omogućava nam postavljanje RIP timer-a u sekundama. 30 (routing update), 90 (invalid timer), 180 (Hold timer), 270 (Flush timer), 360 (sleep timer)
Router(config)#no router rip	Onemogući (zaustavi) RIP routing protocol
Router#debug ip rip	Koristi se za analizu problema. Omogućava nam da vidimo sve RIP vezane aktivnosti u stvarnom vremenu.
Router#show ip rip database	Prikaži RIP bazu podataka uključujući rute

Tablica 4.2. Popis RIP naredbi

5. OSPF (*Open Shortest Path First*) protokol usmjeravanja

RIP protokol ograničen je na maksimum od 15 skokova i namjenjen je za uporabu u manjim mrežama. OSPF protokol napravljen je da udovolji zahtjevima velikih mreža (*enterprise size network*). Kako bi izvršio skaliranje velikih mreža OSPF protokol uvodi **koncept „područja“ (*Area concept*)** koji je vrlo sličan izradi podmreža (*subnetting*). Ovaj koncept omogućava nam da velike mreže podijelimo u više malih mreža koje nazivamo područja ili „*Area*“. Na taj način kontrolira se *broadcasting*, tj. nekontrolirano širenje *broadcast* poruka u ostale dijelove mreže. Zajedno sa *Area* konceptom, OSPF također podržava i *Autonomous System* (AS) koncept koji se koristi u svim računalnim mrežama. Oba koncepta dijele velike mreže u više malih. *Area* koncept je isključivo vezan za OSPF protokol, tj. radi samo uz njega i mi možemo samostalno dodjeljivati adrese *Area* po svojoj želji. Ovdje ćemo se baviti isključivo OSPF usmjeravajućim protokolom i njegovim *Area* konceptom.



Slika 5.1. *AS i Area koncept*

OSPF je ***link-state*** protokol. Što to znači?

Link je veza ili sučelje na kojemu radi OSPF protokol usmjeravanja.

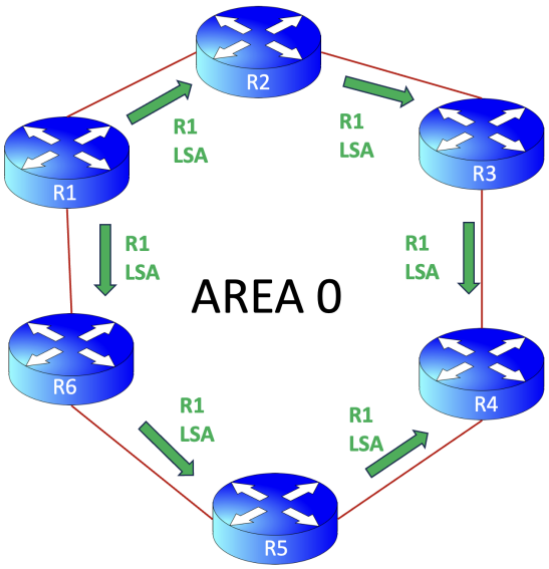
*State (*stanje*)* je skup informacija koje su pridružene svakom sučelju poput IP adrese, *up/down* statusa, mrežne maske, vrste sučelja, vrste mreže, širine pojasa i kašnjenja. OSFP skup tih informacija promatra kao ***state*** nekog linka..

Link state advertisement (LSA) je paket informacija. On sadrži *link-state* i informaciju o usmjeravanju (*routing information*). OSPF koristi LSA pakete za dijeljenje i učenje informacija o mrežnoj topologiji.

Svaki OSPF usmjerivač održava svoju *Link state database* (LSDB). LSDB je skup svih LSA koje je određeni usmjerivač primio. Svaki LSA ima svoj jedinstveni sekvencijski broj (*sequence number*). OSPF sprema LSA u LADB sa tim sekvencijskim brojem.

OSPF usmjerivač generira LSA odmah nakon svog početka rada ili nakon bilo kakvih promjena. Ovaj LSA sadrži kolekciju svih *link-state* ili *link state update* informacija.

Svi usmjerivači međusobno izmjenjuju LSA principom „poplave“. Svaki usmjerivač koji primi LSA paket, spremi kopiju u svoju LSDB bazu, a zatim prosljeđuje LSA drugim usmjerivačima. Slika ispod opisuje taj proces gdje R1 generira LSA i poplavom ga širi do svih usmjerivača kroz mrežu.



Slika 5.2. *Poplavno širenje LSA poruka*

R2 i R5 su prvi koji će zaprimiti ovaj LSA. Oni će osvježiti svoje LSDB baze i nakon toga će LSA proslijediti prema R3 odnosno R6. R3 i R6 će osvježiti svoje baze sa ovim LSA i proslijediti ga dalje do R4. Međutim, samo jedan od usmjerivača ili R3 ili R6 će biti u mogućnosti proslijediti LSA prema R4. Zašto?

Zato što „poplavni“ mehanizam ima zaštitu od nastajanja petlji! Prije slanja LSA prema svom susjedu, svaki usmjerivač prvo pita susjeda „Imaš li LSA sa ovim sekvencijskim brojem?“

Ukoliko susjedni usmjerivač odgovori potvrdno, prosljeđivanje tog LSA biti će zaustavljeno. Iz tog razloga R4 će primiti LSA samo od jednog susjeda; ili od R3 ili od R6.

OSPF usmjerivači razmjenjuju LSA pakete samo sa svojim **susjedima**. Da bi neki usmjerivač postao „susjed“ drugome, moraju se ispuniti određeni uvjeti tj. oni moraju biti u istom području (Area ID), te se moraju poklapati parametri *Authentication, Hello i Dead interval, Stub Area i MTU*. Za ovu razinu znanja razmatrati ćemo samo parametar Area ID koji se treba poklapati.

Mogućnosti i prednosti OSPF protokola

- Podržava IPv4 i IPv6 usmjeravanje.
- Podržava balansiranje opterećenja kod različitih ruta (staza) sa jednakom *cost* vrijednošću za istu destinaciju
- Obzirom da je baziran na otvorenim standardima, raditi će na većini usmjerivača.
- Omogućava topologiju bez petlji zahvaljujući SPF algoritmu.
- On je besklasni (*classless*) protokol.
- Podržava VLSM i sumiranje ruta.
- Podržava neograničen broj skokova.
- Lako skalira velike mreže koristeći koncept „područja“ (*Area concept*).
- Podržava okidno osvježavanje ruta za brzu konvergenciju.

Nedostaci OSPF protokola

- Zahtjeva povećano opterećenje CPU-a usmjerivača zbog SPF algoritma.
- Zahtjeva više RAM-a za pohranu susjedne topologije.
- Kompleksnije je postavljanje i teže je detektirati probleme ukoliko nešto ne radi

Konfiguracija OSPF protokola

Pokretanje OSPF protokola radi se u dva koraka:

- Pokretanje OSPF usmjeravajućeg protokola iz globalnog konfiguracijskog moda
- Definiranje sučelja koja će biti uključena u OSPF

Naredbe za ta dva koraka su:

```
Router(config)# router ospf process_ID
Router(config-router)# network IP_network_# [wild card mask] Area [Area Number]
```

Router(config)# router ospf process ID

Ova naredba pokrenuti će OSPF protokol usmjeravanja na usmjerivaču. Process ID je pozitivni cijeli broj. Možemo koristiti bilo koji broj od 1 do 65,535. Process ID je samo lokalno važan. Možemo imati višestruke OSPF procese na istom usmjerivaču. Process ID nam služi kako bi ih razlikovali. Process ID ne mora se poklapati na svim usmjerivačima.

Router(config-router)# network IP_network_# [wildcard_mask] area [area number]

Ova naredba omogućuje nam da specificiramo sučelja koja želimo uključiti u OSPF proces. Ova naredba traži tri parametra: mrežni broj, *wildcard* masku i *area* broj. Mrežni broj je mrežni ID. Možemo koristiti bilo koju IP adresu uređaja ili mrežnu IP adresu. Na primjer, možemo koristiti 192.168.1.1 (IP adresa uređaja) ili možemo koristiti 192.168.1.0 (mrežna IP adresa). Ako želimo uključiti određeno sučelje onda uglavnom koristimo IP adresu uređaja koja je postavljena na to sučelje. Ukoliko želimo odjednom uključiti više sučelja koje se nalaze na istoj mreži, onda koristimo mrežnu IP adresu i sva sučelja koja pripadaju mreži sa tim mrežnim ID-om biti će uključena u OSPF.

***Wildcard* maska**

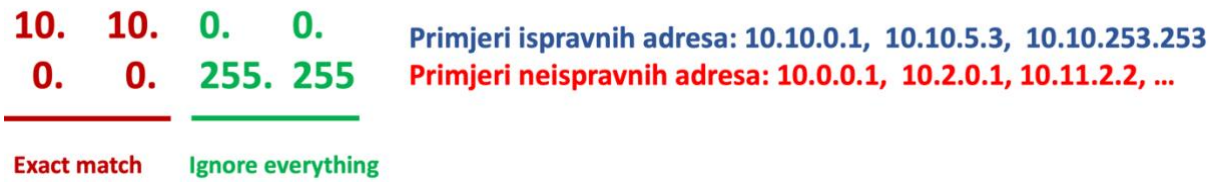
Wildcard maska koristi se zajedno sa mrežnim ID-om kako bi se filtrirala sučelja. *Wildcard* maska nije isto što i maska podmreže (*subnet mask*). *Subnet* maska se koristi za odvajanje mrežnog dijela i dijela za adresiranje uređaja iz IP adrese. Sa druge strane, *wildcard* maska se koristi za pronalazak odgovarajućeg podudaranja okteta u mrežnom dijelu adrese. *Wildcard* maska govori OSPF-u koji dio mrežne adrese se mora podudarati.

Wildcard masku možemo promatrati u decimalnom ili binarnom obliku.Ukoliko je promatramo u decimalnom obliku tada su nam ključne vrijednosti:

0 (*Decimal – octet format*) Wildcard maska nam govori da se odgovarajući oktet u mrežnoj adresi mora u potpunosti podudarati.

255 (*Decimal – octet format*) Wildcard maska nam govori da ne moramo brinuti o podudaranju odgovarajućeg okteta u mrežnoj adresi.

Primjer:

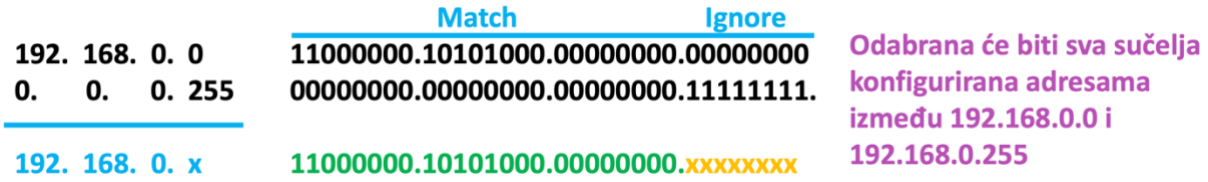


***Slika 5.3.** OSPF Wilecard maska primjer 1*

Ukoliko pak promatramo mrežnu masku u binarnom obliku tada su nam ključne vrijednosti:

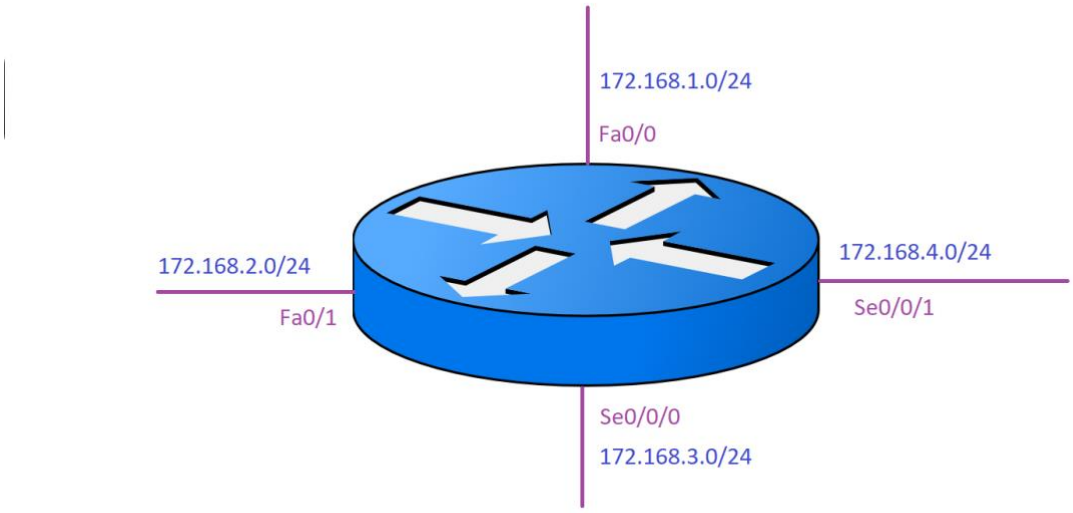
0 (*Binary – bit format*)*Wildcard* maska nam govori da se odgovarajući oktet u mrežnoj adresi mora u potpunosti podudarati.

255 (*Binary – bit format*)*Wildcard* maska nam govori da ne moramo brinuti o podudaranju odgovarajućeg okteta u mrežnoj adresi.



Slika 5.4. *OSPF Wilecard maska primjer 2*

OSPF je besklasni (*classless*) protokol. Uz pomoć *wildcard* maske također možemo filtrirati i podmreže. Npr. pogledajmo sljedeću sliku



Slika 5.5. *OSPF primjer*

Imamo četri mreže:

- 172.168.1.0/24,
- 172.168.2.0/24,
- 172.168.3.0/24 i
- 172.168.4.0/24

koje su dobivene *subnetiranjem* iz jedne mreže klase B 172.168.0.0/16.

Ukoliko koristimo konfiguraciju po klasama, ona ne razumije koncept izrade podmreža. Klasne konfiguracije rade isključivo sa *default* maskama. *Default* maska za ovu mrežu klase B ima 16 bita, pa će klasni protokol pronaći samo preklapanje prvih 16 bita (172.168.x.y) mrežne adrese. Klasni protokol usmjeravanja poput RIP protokola ne može napraviti razliku između podmreža.

Bezklasni protokol usmjeravanja poput OSPF-a može bez problema filtrirati i podmreže. Korištenjem *wildcard* maske više nemamo ograničenja sa definiranim granicama mreža.

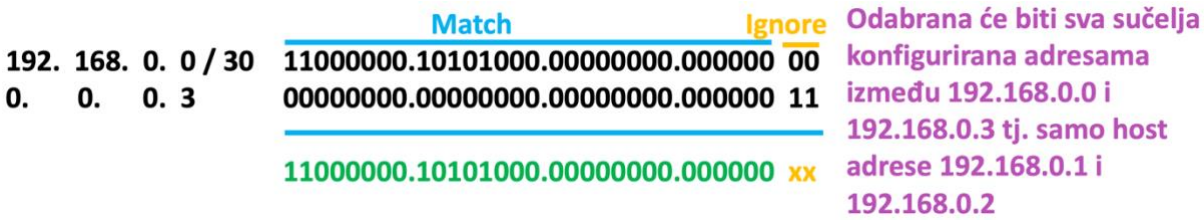
Npr. Želimo isključiti serijsko sučelje iz gornje konfiguracije. Možemo koristiti *wildcard* masku 0.0.0.255 kako bi pogodili masku podmreže /24.

```
Router(config-router)# network 172.168.1.0 0.0.0.255
Router(config-router)# network 172.168.2.0 0.0.0.255
```

Gore navedene naredbe će tražiti od usmjerivača da se izvrši preklapanje /24 bita adrese umjesto *default*-nih /16 bita. Sada će usmjerivač gledati samo 172.168.1.x i 172.168.2.x mreže. Naša serijska sučelja pripadaju 172.168.3.0/24 i 172.168.4.0/24 mrežama, te ne upadaju u ove kriterije.

Idemo pokazati još jedan primjer. Ako koristimo slijedeće naredbe, koje sučelje će biti uključeno u OSPF?

```
Router(config-router)# network 192.168.0.0 0.0.0.3
```

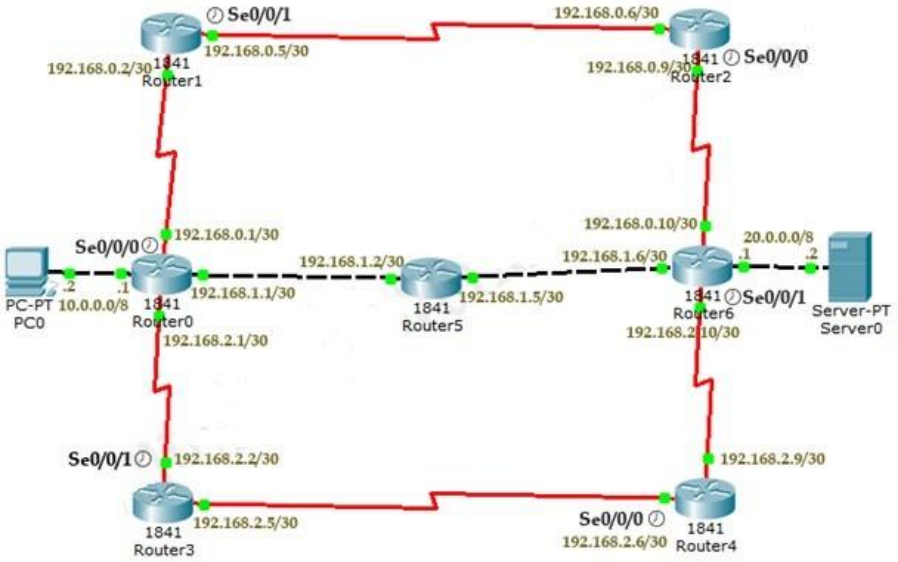


Slika 5.6. *Wilecard maska primjer 3*

U ovom slučaju važeće su samo adrese uređaja 192.168.0.1 i 192.168.0.2. Znači svaki uređaj koji ima te adrese će biti odabran. /30 mreža se obično koristi za serijske veze koje trebaju samo dvije adrese za uređaje, po jednu za svaki kraj serijskog kabela.

Idemo sada pogledati značenje trećeg parametra koje koristimo kod aktivacije OSPF protokola, a to je Area ID. Ovaj parametar kaže usmjerivaču da ona sučelja za koje su se predhodni uvjeti poklopili, uvrsti u specifično područje (*Area*).

Zadatak – Konfiguracija OSPF usmjerivačkog protokola



Slika 5.7. *Topologija OSPF zadatka*

Uređaj	**Sučelje**	**IP adresa**	**Spojen sa**
PC0	Fa0/0	10.0.0.2/8	Router0 Fa0/0
Router0	Fa0/0	10.0.0.1/8	PC0 Fa0/0
Router0	Fa0/1	192.168.1.1/30	Router5 Fa0/1
Router5	Fa0/1	192.168.1.2/30	Router0 Fa0/1
Router5	Fa0/0	192.168.1.5/30	Router6 F0/0
Router6	Fa0/0	192.168.1.6/30	Router5 Fa0/0
Router6	Fa0/1	20.0.0.1/8	Server0 Fa0/0
Server0	Fa0/0	20.0.0.2/8	Router6 Fa0/1
Router0	Serial 0/0/0 (DCE)	192.168.0.1/30	Router1 Se0/0/0
Router1	Serial 0/0/0	192.168.0.2/30	Router0 Se0/0/0
Router1	Serial 0/0/1 (DCE)	192.168.0.5/30	Router2 Se0/0/1
Router2	Serial0/0/1	192.168.0.6/30	Router1 Se0/0/1
Router2	Serial 0/0/0 (DCE)	192.168.0.9/30	Router6 Se0/0/0
Router6	Serial 0/0/0	192.168.0.10/30	Router2 Se0/0/0
Router0	Serial 0/0/1	192.168.2.1/30	Router3 Se0/0/1
Router3	Serial 0/0/1 (DCE)	192.168.2.2/30	Router0 Se0/0/1
Router3	Serial 0/0/0	192.168.2.5/30	Router4 Se0/0/0
Router4	Serial 0/0/0 (DCE)	192.68.2.6/30	Router3 Se0/0/0
Router4	Serial 0/0/1	192.168.2.9/30	Router6 Se0/0/1
Router6	Serial0/0/1 (DCE)	192.168.2.10/30	Router4 Se0/0/1

Tablica 5.1. Povezivanje sučelja u zadatku

Korak 1. Dodjeli IP adrese računalu (PC) i serveru

Dvostrukim „klikom“ miša otvori PC0, idi na **Desktop** meni i odaberi **IP Configuration**. Dodjeli računalu IP adresu 10.0.0.2/8 . Isti postupak ponovi i za poslužitelj (*server*), ali njemu dodijeli IP adresu 20.0.0.2/8.

Korak 2. Dodjeli IP adrese sučeljima svih usmjerivača

Dvostruki klik mišom na **Router0,** klik na **CLI,** te pritisnite „Enter“ tipku kako bi ušli u *command prompt* sučelje od usmjerivača **Router0**. Trebamo konfigurirati četri sučelja na ovom usmjerivaču - FastEthernet0/0, FastEthernet0/1, Serial 0/0/0 i Serial0/0/1. Po tvorničkim postavkama (*default*) sva sučelja na usmjerivaču su administrativno u isključenom stanju (*administratively down*) za vrijeme uključenja usmjerivača. Moramo im konfigurirati IP adrese i ostale parametre prije nego što ih možemo upotrebiti sa usmjeravanje prometa. Postavljanje IP adrese nekom sučelju obavlja se iz tzv. *Interface* moda u koji prvo moramo ući. *Interface* modu pristupa se iz globalnog konfiguracijskog moda. Slijedeće naredbe koriste se za ulazak u globalni konfiguracijski mod.

```
Router>enable
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

Iz globalnog konfiguracijskog moda možemo ući u *interface* mod, a iz njega pak konfigurirati sve parametre tog sučelja. Slijedeće naredbe koriste se za postavljanje IP adrese na FastEthernet0/0 i FastEthernet0/1.

```
Router(config)#interface fastEthernet 0/0 –Naredba za ulaz u interface mod
Router(config-if)#ip address 10.0.0.1 255.0.0.0 - Naredba za postavljanje IP adrese sučelju
Router(config-if)#no shutdown - Naredba za „podizanje“ sučelja u radno stanje
Router(config-if)#exit - Naredba za povratak u globalni konfiguracijski mod
Router(config)#interface fastEthernet 0/1
Router(config-if)#ip address 192.168.1.1 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
```

Serijska veza treba definiciju još dva dodatna parametra ***clock rate*** i ***bandwidth*** na DCE kraju veze. O tome smo detaljno pričali u prošloj vježbi pa ovdje nećemo to ponovno objašnjavati.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 192.168.0.1  255.255.255.252
Router(config-if)#clock rate 64000
Router(config-if)#bandwidth 64
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/1
Router(config-if)#ip address 192.168.2.1  255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
```

Iste naredbe koristiti ćemo i za konfiguraciju ustalih usmjerivača.

Router1

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 192.168.0.2  255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 192.168.0.5  255.255.255.252
Router(config-if)#clock rate 64000
Router(config-if)#bandwidth 64
Router(config-if)#no shutdown
Router(config-if)#exit
```

Router2

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 192.168.0.9  255.255.255.252
Router(config-if)#clock rate 64000
Router(config-if)#bandwidth 64
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/1
Router(config-if)#ip address 192.168.0.6  255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
```

Serijsko sučelje ima *default* širinu pojasa od 1544Kbps. Ukoliko mi ne odredimo neku drugu širinu pojasa, usmjerivač će koristiti *default* vrijednost. Da bi vidjeli kako to radi na ostalim usmjerivačima nećemo kreirati ovaj parametar.

Router6

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 192.168.0.10  255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/1
Router(config-if)#ip address 192.168.2.10  255.255.255.252
Router(config-if)#clock rate 64000
Router(config-if)#bandwidth 64
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 192.168.1.6 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface fastethernet 0/1
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
```

Router5

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 192.168.1.5 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface fastethernet 0/1
Router(config-if)#ip address 192.168.1.2 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
```

Router3

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 192.168.2.5  255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/1
Router(config-if)#ip address 192.168.2.2  255.255.255.252
Router(config-if)#clock rate 64000
Router(config-if)#bandwidth 64
Router(config-if)#no shutdown
Router(config-if)#exit
```

Router4

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 192.168.2.6  255.255.255.252
Router(config-if)#clock rate 64000
Router(config-if)#bandwidth 64
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/1
Router(config-if)#ip address 192.168.2.9  255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
```

Sada svi usmjerivači imaju postavljana sva sučelja. Oni neće početi razmjenjivati informacije o tome prije nego mi pokrenemo neki od protokola usmjeravanja, u ovom slučaju OSPF.

Korak 3. Postavljanje OSPF priotokola usmjeravanja

Router0

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 10
Router(config-router)#network 10.0.0.0 0.255.255.255 area 0
Router(config-router)#network 192.168.0.0 0.0.0.3 area 0
Router(config-router)#network 192.168.1.0 0.0.0.3 area 0
```

```
Router(config-router)#network 192.168.2.0 0.0.0.3 area 0
Router(config-router)#exit
Router(config)#
```

Router1

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 10
Router(config-router)#network 192.168.0.0 0.0.0.3 area 0
Router(config-router)#network 192.168.0.4 0.0.0.3 area 0
Router(config-router)#exit
Router(config)#
```

Router2

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 20
Router(config-router)#network 192.168.0.4 0.0.0.3 area 0
Router(config-router)#network 192.168.0.8 0.0.0.3 area 0
Router(config-router)#exit
Router(config)#
```

Router6

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 50
Router(config-router)#network 192.168.0.8 0.0.0.3 area 0
Router(config-router)#network 192.168.2.8 0.0.0.3 area 0
Router(config-router)#network 192.168.1.4 0.0.0.3 area 0
Router(config-router)#exit
Router(config)#
```

Router5

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 80
Router(config-router)#network 192.168.1.0 0.0.0.3 area 0
Router(config-router)#network 192.168.1.4 0.0.0.3 area 0
Router(config-router)#exit
Router(config)#
```

Router4

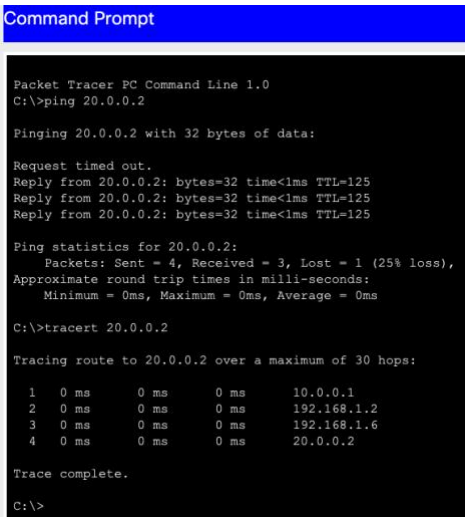
```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 40
Router(config-router)#network 192.168.2.8 0.0.0.3 area 0
Router(config-router)#network 192.168.2.4 0.0.0.3 area 0
Router(config-router)#exit
Router(config)#
```

Router3

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 30
Router(config-router)#network 192.168.2.0 0.0.0.3 area 0
```

```
Router(config-router)#network 192.168.2.4 0.0.0.3 area 0
Router(config-router)#exit
Router(config)#
```

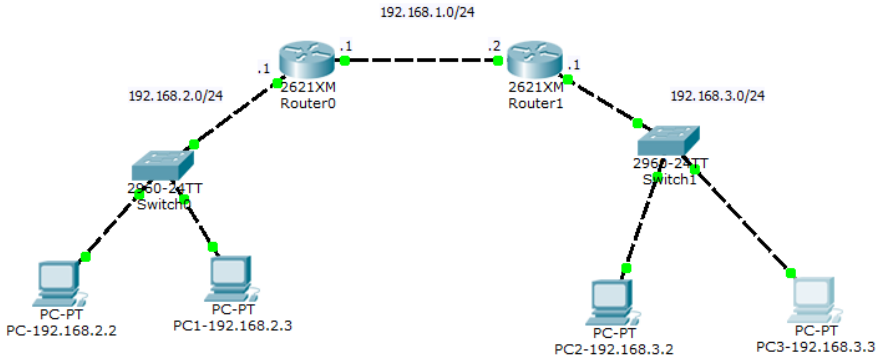
Naša mreža sada je spremna za OSPF usmjeravanje. Da bi provjerili da li smo ispravno izvršili konfiguraciju koristiti ćemo ***ping*** naredbu. Postoje dvije rute između PC i server uređaja. Da bi vidjeli koja od njih je u funkciji možemo koristiti naredbu ***tracert.*** Pristupimo *command prompt* sučelju na PC1 i koristimo ***ping*** naredbu za testiranje veze sa Server0. Nakon toga koristimo ***tracert*** naredbu da vidimo koja ruta se koristi.



Slika 5.8. *Prikaz rezultata nakon uporabe* ***tracert*** *naredbe na PC0*

Samostalni zadatak

Pokušajte samostalno riješiti slijedeći zadatak, bez gledanja rješenja koje se nalazi na slijedećoj stranici



Slika 5.9. *OSPF - topologija mreže samostalnog zadatka*

1. Nacrtati topologiju kao na slici
2. Kreirati sučelja na usmjerivačima i pridjeliti im IP adrese
3. Postaviti ip adresu, subnet masku i default gateway na računalima
4. Oglasiti OSPF
5. Provjeriti rad naredbom ***ping*** iz jedne mreže u drugu

Rješenje:

Router 0

```
Router>enable
Router#configure terminal
Router(config)#hostname R0      -mjenjamo ime usmjerivača u R0
R0(config)#interface fastEthernet 0/1
R0(config-if)#ip address 192.168.2.1 255.255.255.0    -dodjeljuje ip adresu sučelju 0/1
R0(config-if)#no shutdown
R0(config-if)#exit
R0(config)#interface fastEthernet 0/0
R0(config-if)#ip address 192.168.1.1 255.255.255.0     -dodjeljuje ip adresu sučelju 0/0
R0(config-if)#no shutdown
R0(config-if)#exit
R0(config)#
```

Router1

```
Router>enable
Router(config)#hostname R1
R1(config)#interface fastEthernet 0/1
R1(config-if)#ip address 192.168.3.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#exit

R1(config)#interface fastEthernet 0/0
R1(config-if)#ip address 192.168.1.2 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#
```

OSPF - konfiguracija

RO

```
R0(config)#router ospf 1
R0(config-router)#network 192.168.2.0 0.0.0.255 area 0
R0(config-router)#network 192.168.1.0 0.0.0.255 area 0
R0(config-router)#network 192.168.3.0 0.0.0.255 area 0
R0(config-router)#exit
R0(config)#
```

R1

```
R1(config)#router ospf 1
R1(config-router)#network 192.168.2.0 0.0.0.255 area 0
R1(config-router)#network 192.168.1.0 0.0.0.255 area 0
R1(config-router)#network 192.168.3.0 0.0.0.255 area 0
R1(config-router)#exit
R1(config)#
```

Router#**show ip protocols** - prikazuje protokol koji se koristi.

Router#**show ip route** - prikazuje ip rutu.

```
R0#show ip protocols

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 192.168.2.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    192.168.2.0 0.0.0.255 area 0
    192.168.1.0 0.0.0.255 area 0
    192.168.3.0 0.0.0.255 area 0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.2.1           110      00:02:49
    192.168.3.1           110      00:02:49
  Distance: (default is 110)

R0#
```

Slika 5.10. *Prikaz odgovora nakon korištenja* ***show ip protocols*** *naredbe*

```
R0#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.1.0/24 is directly connected, FastEthernet0/0
C    192.168.2.0/24 is directly connected, FastEthernet0/1
O    192.168.3.0/24 [110/2] via 192.168.1.2, 00:01:39, FastEthernet0/0
R0#
```

Slika 5.11. *Prikaz odgovora nakon korištenja* ***show ip route*** *naredbe*

6. Statičko „*default*“ usmjeravanje

U slučajevima gdje je iz nekog područja (mreže) potrebno omogućiti izlaz paketa prema području o kojemu se ništa nezna, koristi se tzv. „*default*“ statička ruta. Statička *default* ruta još se naziva i „*last resort gateway*“ ruta jer usmjerivaču definira na koje sučelje treba proslijediti neki IP paket ukoliko u svojoj tablici usmjeravanja nije u mogućnosti pronaći unos za tu mrežu. Kako izgleda konfiguracija statičke *default* rute:

```
Router(config)# ip route 0.0.0.0 0.0.0.0 IP adresa slijedećeg sučelja

                                    ili

Router(config)#  ip  route  0.0.0.0  0.0.0.0  sučelje  kroz  koje  želimo
proslijediti paket (npr. serial2/0)
```

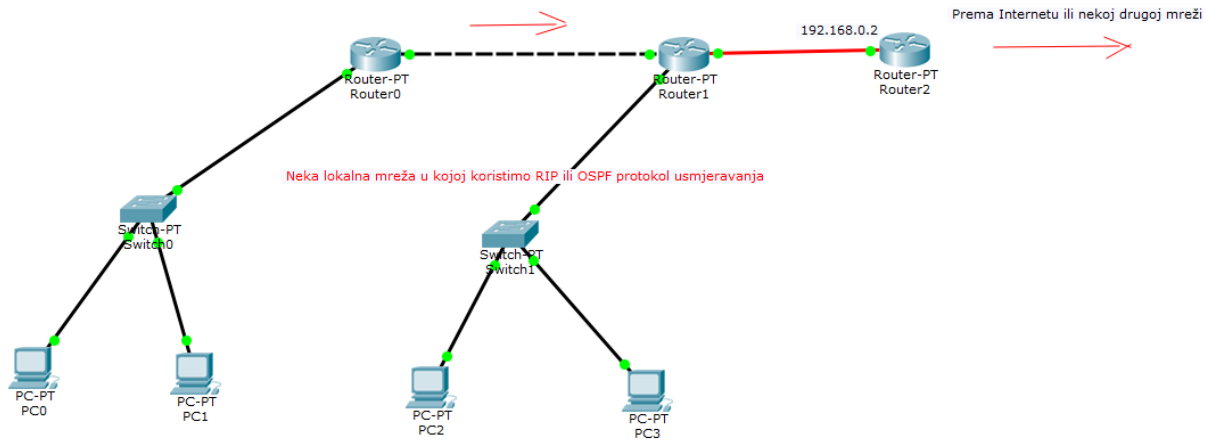
Opis: Ukoliko na usmjerivač stigne IP paket, a u tablici usmjeravanja ne postoji unos koji pokazuje put prema njegovoj mreži, tada će paket biti proslijeđen *default* rutom. Ako *default* ruta nije definirana, paket će jednostavno biti odbačen.

Primjer:

Router1

```
Router(config)#ip  route 0.0.0.0 0.0.0.0 192.168.0.2
```

Značenje ove naredbe moglo bi se objasniti riječima „*Ukoliko usmjerivač zaprimi paket kojemu je odredišna adresa u mreži o kojoj usmjerivač nema podataka u svojoj tablici usmjeravanja, tj. nezna gdje bi trebao preusmjeriti paket, ...tada ga treba proslijediti ka sučelju 192.168.0.2*“



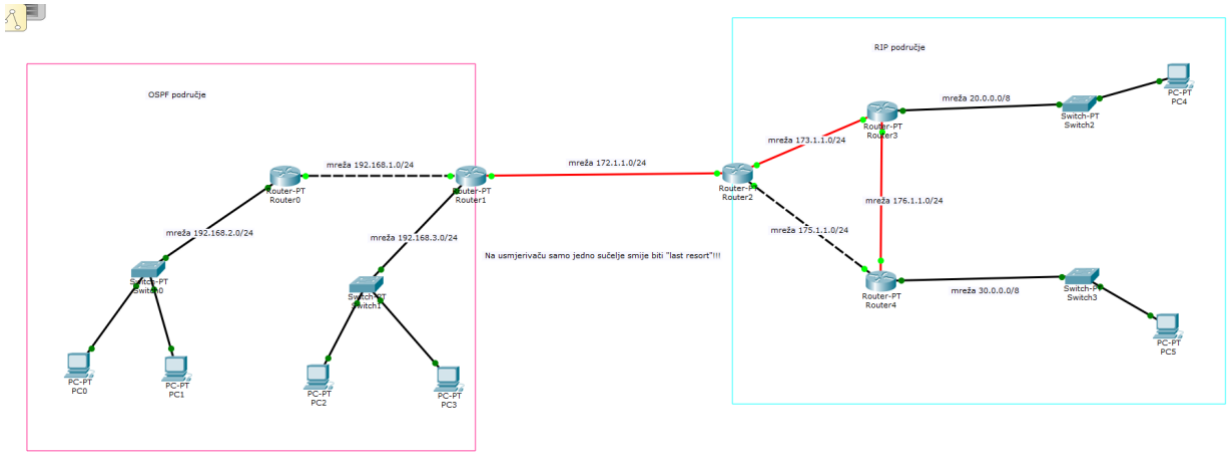
Slika 6.1. *Mrežna topologija za navedeni primjer*

U slučaju da ima više usmjerivača unutar mreže, kao na slici iznad, potrebno je svakom usmjerivaču definirati statičku „*default*“ rutu, tj. ukoliko neko računalo spojeno na Router 0 želi poslati paket nekoj usmjerivaču nepoznatoj mreži, treba mu reći da takav paket proslijedi na ulazno sučelje usmjerivača Router 1. Kako ni usmjerivač Router1 također nezna put do te nepoznate mreže, on će taj paket proslijediti također po „*default*“ ruti prema usmjerivaču Router2. Obzirom da se „*default*“ ruta postavlja kao statička, podaci o njoj se ne razmjenjuju

putem usmjeravajućih protokola. Primjer uporabe takvog usmjeravanja nalazimo i pri spajanju dvaju područja u kojima se koriste različiti protokoli usmjeravanja, pa također nema razmjene znanja o mrežama koje se nalaze unutar jednog ili drugog područja.

Zadatak 6.1 – Povezivanje OSPF i RIP područja „*default*“ statičkom rutom

Izvrši povezivanje dvaju područja sa slike ispod u kojima se primjenjuju različiti protokoli usmjeravnja primjenom statičke „default“ rute.



Slika 6.2. *Povezivanje RIP i OSPF područja*

Treba paziti da se omogući obostrana komunikacija, te je „*default*“ rute potrebno definirati na svim usmjerivačima s jedne i druge strane.

Sada u CPT otvorite vježbu „Povezivanje RIP i OSPF područja pomoću *last resort* rute“ (nalazi se na Moodle-u)

- Izvršite konfiguraciju svih sučelja po datim adresama
- U lijevoj mreži pokrenite OSPF protokol, a u desnoj RIP. Provjerite rad.
- Sada konfigurirajte „*last resort*“ statičke rute na usmjerivačima kako bi omogućili komunikaciju između OSPF i RIP područja.

Rješenje:

Router0

```
Router>enable
Router#conf t
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 192.168.2.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#router ospf 10
```

```
Router(config-router)#network 192.168.1.0 0.0.0.255 area 0
Router(config-router)#network 192.168.2.0 0.0.0.255 area 0
Router(config-router)#exit
Router(config)#ip route 0.0.0.0 0.0.0.0 192.168.1.2
```

Router1

```
Router>enable
Router#conf t
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 192.168.1.2 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#ip address 192.168.3.1 255.255.255.0
Router(config-if)#exit
Router(config)#interface FastEthernet4/0
Router(config-if)#ip address 172.1.1.1 255.255.255.0
Router(config-if)#exit
Router(config)#router ospf 10
Router(config-router)#network 192.168.1.0 0.0.0.255 area 0
Router(config-router)#network 192.168.3.0 0.0.0.255 area 0
Router(config-router)#exit
Router(config)#ip route 0.0.0.0 0.0.0.0 172.1.1.2
```

Router2

```
Router>enable
Router#conf t
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 175.1.1.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface FastEthernet4/0
Router(config-if)#ip address 172.1.1.2 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface FastEthernet5/0
Router(config-if)#ip address 173.1.1.2 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 172.1.0.0
Router(config-router)#network 173.1.0.0
Router(config-router)#network 175.1.0.0
Router(config-router)#exit
Router(config)#ip route 0.0.0.0 0.0.0.0 172.1.1.1
```

Router3

```
Router>enable
Router#conf t
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
```

```
Router(config)#interface FastEthernet4/0
Router(config-if)#ip address 176.1.1.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface FastEthernet5/0
Router(config-if)#ip address 173.1.1.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 20.0.0.0
Router(config-router)#network 173.1.0.0
Router(config-router)#network 176.1.0.0
Router(config-router)#exit
Router(config)#ip route 0.0.0.0 0.0.0.0 173.1.1.2
```

Router4

```
Router>enable
Router#conf t
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 175.1.1.2 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface FastEthernet4/0
Router(config-if)#ip address 176.1.1.2 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 30.0.0.0
Router(config-router)#network 175.1.0.0
Router(config-router)#network 176.1.0.0
Router(config-router)#exit
Router(config)#ip route 0.0.0.0 0.0.0.0 175.1.1.1
```

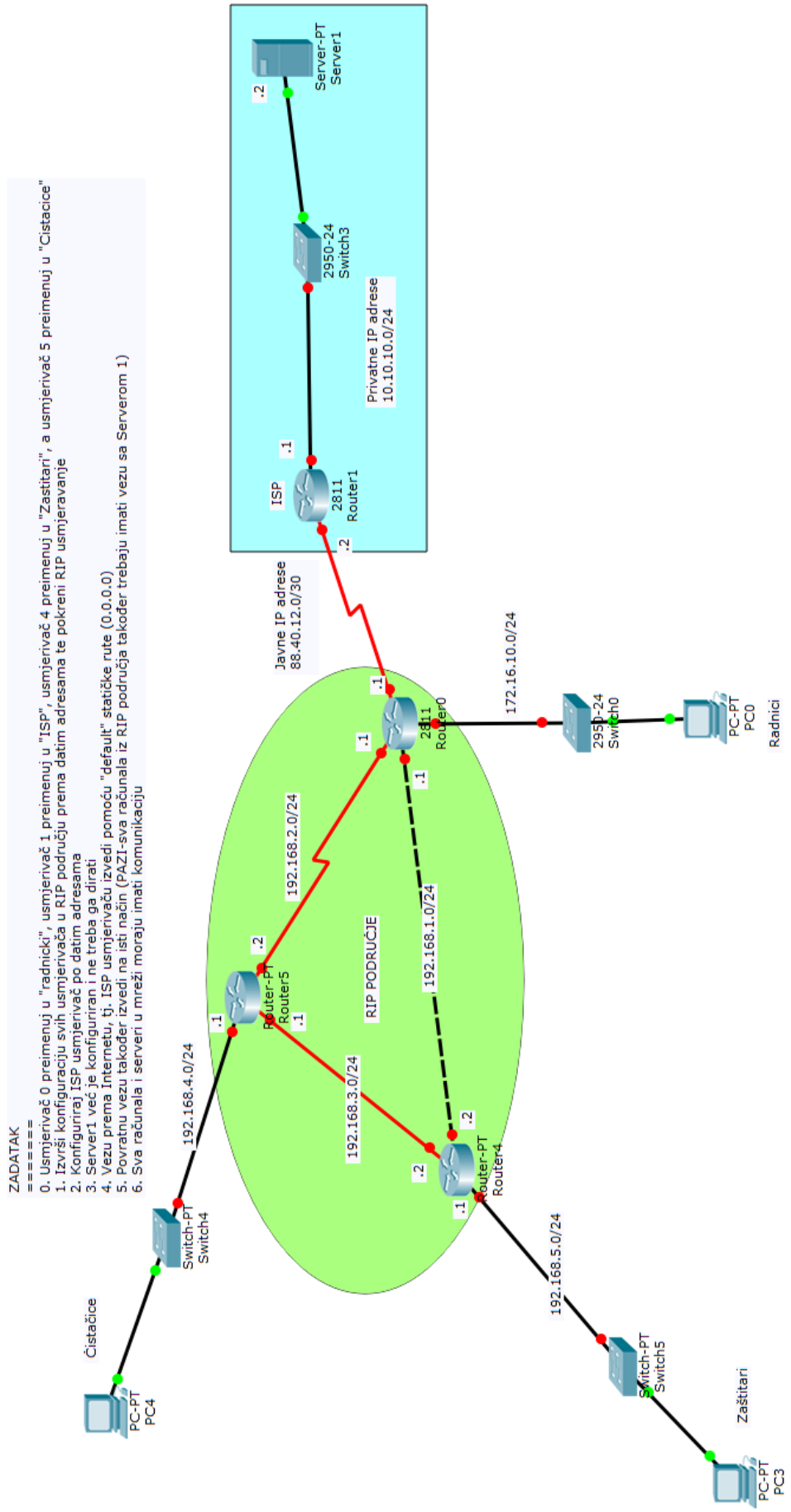
Zadatak 6.2 – Povezivanje sa mrežom ISP-a pomoću statičkog „*Default*“ usmjeravanja

U računalnim mrežama postoje dva uobičajena načina kako spajamo naše „privatne“ mreže na Internet tj. na mrežu davatelja usluga (ISP)

1. Definicijom tzv. *default* statičke rute na vašem usmjerivaču koji vas spaja sa mrežom ISP-a
2. Pokretanje BGP (*Boarder Gateway Protocol*) protokola na vašem usmjerivaču koji vas spaja sa ISP-om. Usmjeravanje Internetom bazirano je na BGP protokolu. Međutim usmjerivači koji su u stanju koristiti BGP protokol su vrlo skupi jer BGP tablice usmjeravanja pohranjuju rute prema svim mrežama na Internetu.

Zadatak:

U Cisco Packet Tracer aplikaciji pokrenite vježbu imena „Statička default ruta“ (nalazi se na Moodle-u), izvršite konfiguraciju svih usmjerivača unutar „privatne“ mreže, pokrenite RIP usmjeravanje, te pomoću *default* statičke rute izvršite povezivanje sa ISP usmjerivačem. Na ISP usmjerivaču na isti način uspostavite vezu prema „privatnoj“ mreži kako bi se mogla izvršiti „ping“ naredba za provjeru veze sa poslužiteljom.



Slika 6.3. *Topologija mreže za zadatak 6.2*

Idemo prvo promjeniti ime te izvršiti IP konfigurirati sučelja na svim usmjerivačima.

Router 0

```
Router>enable
Router#conf t
Router(config)#hostname Radnicki        - promjena imena
Radnicki(config)#interface fa 0/0
Radnicki(config-if)#ip address 172.16.10.1 255.255.255.0
Radnicki(config-if)#no shut
Radnicki(config-if)#exit
Radnicki(config)#interface fa 0/1
Radnicki(config-if)#ip address 192.168.1.1 255.255.255.0
Radnicki(config-if)#no shut
Radnicki(config-if)#exit
Radnicki(config)#interface serial 0/0/1
Radnicki(config-if)#ip address 192.168.2.1 255.255.255.0
Radnicki(config-if)#no shut
Radnicki(config-if)#exit
Radnicki(config)#interface serial 0/0/0
Radnicki(config-if)#ip address 88.40.12.1 255.255.255.252
Radnicki(config-if)#clock rate 64000
Radnicki(config-if)#bandwidth 64
Radnicki(config-if)#no shut
```

Router 1 (ISP)

```
Router>enable
Router#conf t
Router(config)#hostname ISP
ISP(config)#interface serial 0/0/0
ISP(config-if)#ip address 88.40.12.2 255.255.255.252
ISP(config-if)#no shut
ISP(config-if)#exit
ISP(config)#interface fa 0/0
ISP(config-if)#ip address 10.10.10.1 255.255.255.0
ISP(config-if)#no shut
```

Router 4

```
Router>enable
Router#conf t
Router(config)#hostname Zastitari
Zastitari(config)#interface fa 0/0
Zastitari(config-if)#ip address 192.168.5.1 255.255.255.0
Zastitari(config-if)#no shut
Zastitari(config-if)#exit
Zastitari(config)#
Zastitari(config)#interface fa 1/0
Zastitari(config-if)#ip address 192.168.1.2 255.255.255.0
Zastitari(config-if)#no shut
Zastitari(config-if)#exit
Zastitari(config)#interface fa 4/0
Zastitari(config-if)#ip address 192.168.3.2 255.255.255.0
Zastitari(config-if)#no shut
Zastitari(config-if)#exit
```

Router 5

```
Router>enable
```

```
Router#conf t
Router(config)#hostname Cistacice
Cistacice(config)#interface fa 0/0
Cistacice(config-if)#ip address 192.168.4.1 255.255.255.0
Cistacice(config-if)#no shut
Cistacice(config-if)#exit
Cistacice(config)#interface fa 4/0
Cistacice(config-if)#ip address 192.168.3.1 255.255.255.0
Cistacice(config-if)#no shut
Cistacice(config-if)#exit
Cistacice(config)#interface serial 2/0
Cistacice(config-if)#ip address 192.168.2.2 255.255.255.252
Cistacice(config-if)#clock rate 64000
Cistacice(config-if)#bandwidth 64
Cistacice(config-if)#no shut
Cistacice(config-if)#exit
```

Sad krećemo na pokretanje RIP protokola na svakom od usmjerivača i postaviti *default* rute

Router 0 (Radnicki)

```
Radnicki>enable
Radnicki#conf t
Radnicki(config)#router rip     - pokretanje RIP protokola
Radnicki(config-router)#network 172.16.10.0
Radnicki(config-router)#network 192.168.1.0
Radnicki(config-router)#network 192.168.2.0
Radnicki(config-router)#exit
Radnicki(config)#ip route 0.0.0.0 0.0.0.0 88.40.12.2    – postavljanje default rute
```

Router 1 (ISP) – PAZI!!! On nije u RIP području!!!

```
ISP>enable
ISP#conf t
ISP(config)#ip route 0.0.0.0 0.0.0.0 88.40.12.1        – postavljanje default rute
```

Router 4

```
Zastitari>enable
Zastitari#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Zastitari(config)#router rip
Zastitari(config-router)#network 192.168.1.0
Zastitari(config-router)#network 192.168.3.0
Zastitari(config-router)#network 192.168.5.0
Zastitari(config-router)#exit
Zastitari(config)#ip route 0.0.0.0 0.0.0.0 192.168.1.1 – postavljanje default rute
```

Router 5

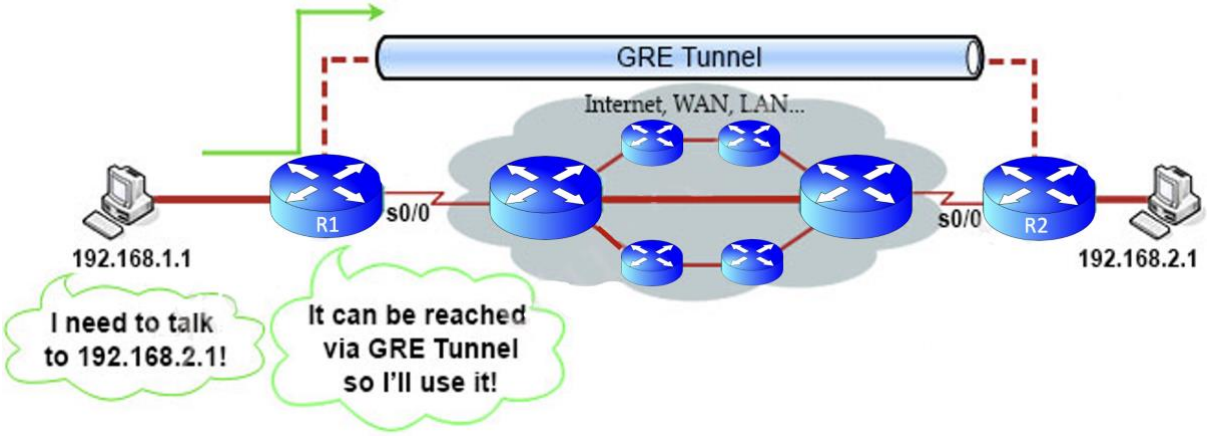
```
Cistacice>enable
Cistacice#conf t
Cistacice(config)#router rip
Cistacice(config-router)#network 192.168.4.0
Cistacice(config-router)#network 192.168.3.0
Cistacice(config-router)#network 192.168.2.0
Cistacice(config-router)#exit
Cistacice(config)#ip route 0.0.0.0 0.0.0.0 192.168.2.1– postavljanje default rute
```

Sada treba postaviti IP konfiguracije na sva racunala u mreži i provjeriti funkcionalnost.

7. GRE kroz IP VPN tunele

Što su to GRE VPN tuneli?

VPN tuneli nam omogućavaju povezivanje geografski razdvojenih privatnih LAN mreža kroz javnu infrastrukturu (Internet). Privatne lokalne mreže spojene pomoću tunela preko Interneta imaju potpunu transparentnost jedne prema drugima, te mogu u potpunosti koristiti sve resurse iz bilo koje zasebne LAN mreže kao da su sve međusobno lokano povezane. Kod VPN tunelski povezanih privatnih mreža, mreže međusobno mogu u potpunosti komunicirati jer su zaglavlja IP paketa (koji sadrže privatne IP adrese) skrivena za javnu infrastrukturu tj. za javne internet usmjerivače. Usmjerivači javne mreže (Interneta) nemaju saznanja o tome da privatne mreže međusobno komuniciraju preko njih. Za razliku od IPSec ili OpenVPN tunela, ***Generic Routing Encapsulation*** **(GRE)** tuneli **ne pružaju** sigurnosne usluge, (autentifikacija) niti enkripciju. To je osnovni razlog zašto se GRE VPN tuneli ne preporučuju kao solucija za povezivanje udaljenih mreža kod kojih sigurnost i povjerljivost imaju veliku ulogu.

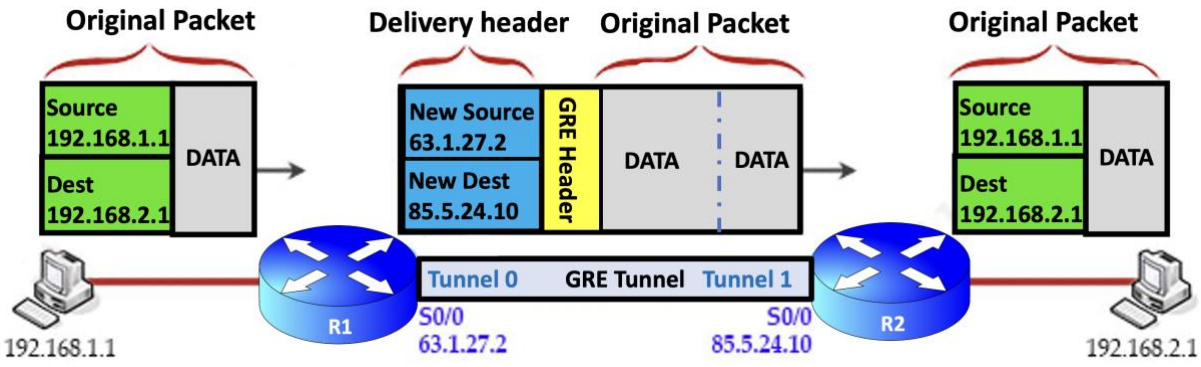


Slika 7.1. *GRE tunel [7]*

Uzmimo da su R1 i R2 usmjerivači u udaljenim uredima jedne firme, a iza sebe imaju spojene uredske LAN mreže. Iako R1 i R2 nisu direkno međusobno spojeni, svim računalima unutar LAN mreža izgleda kao da jesu jer između njih koriste GRE tunel. Ukoliko pogledamo tablice usmjeravanja u R1 i R2 vidjeti ćemo da oni stvarno i jesu povezani.

Kako funkcioniraju GRE tuneli?

Kada izvorišni usmjerivač šalje IP paket u GRE tunel on ga jednostavno cijelog „zapakira“ u jedan veći paket sa dva zaglavlja. Prvo zaglavlje je **GRE zaglavlje** koje služi za upravljanje samim tunelom, a drugo zaglavlje je tzv. „dostavno zaglavlje“ (“*Delivery header*”) koje uključuje novu izvorišnu i odredišnu virtualnu IP adresu tunela. Ovaj proces nazivamo enkapsulacija.



Slika 7.2. *GRE enkapsulacija [7]*

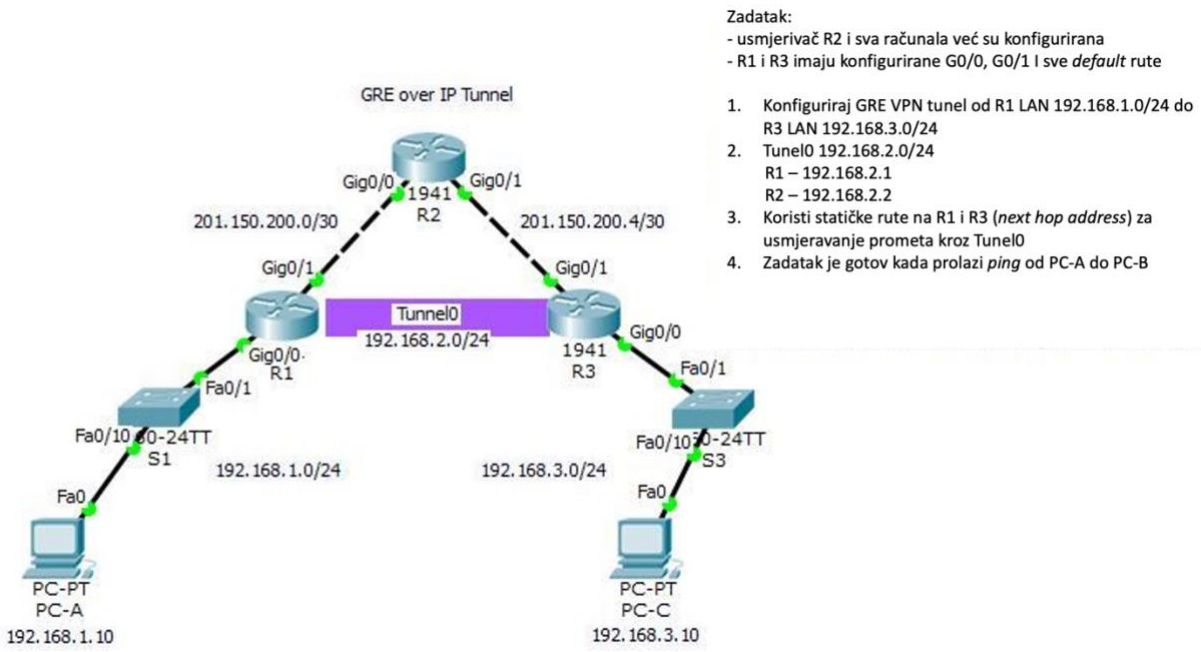
Kada R1 primi IP paket iz svoje LAN mreže, ubaci ga u novi paket sa GRE i „dostavnim“ zaglavljem. Dostavno zaglavlje novu izvorišnu IP adresu 63.1.27.2 (IP adresa fizičkog sučelja R1 usmjerivača koje se koristi kao ulaz/izlaz iz tunela) i novu odredišnu adresu 85.5.24.10 (IP adresa fizičkog sučelja R2 usmjerivača koje se koristi za ulaz/izlaz iz tog tunela).

Kada GRE paket stigne na drugi kraj tunela (u ovom slučaju na usmjerivač R2), prijamni usmjerivač skida GRE i „dostavno“ zaglavlje, te se paket vraća u svoj originalni oblik.

NAPOMENA: U praksi možemo koristiti „*loopback interface*“ kao ulaz/izlaz iz tunela što će nam omogućiti da promet prolazi trenutno najboljom trasom.

Zadatak: Kreiranje GRE kroz IP VPN tunel

Zadana je mrežna topologija kao na slici ispod.



Slika 7.3. *Mrežna topologija za zadatak 7.1[7]*

Kako se konfiguriraju VPN tuneli?

1. Sučelje usmjerivača koji povezuje neku LAN mrežu sa javnom infrastrukturom ima javnu IP adresu jer sa druge strane te veze stoji usmjerivač javnog pružatelja usluga (ISP) koji je također na javnim IP adresama.
2. Obzirom da želimo imati potpuno transparentnu povezanost naših LAN mreža putem privatnih adresa, potrebno je na tom istom sučelje kreirati i „virtualnu“ privatnu IP adresu kako bi naš LAN (a i onaj na drugoj strani kojeg želimo spojiti) mislili da su direkno spojeni.
3. Kako bismo to mogli ostvariti potrebno je kreirati tunel, odrediti koje sučelje je ulaz u taj tunel, dodjeliti „virtualnu“ privatnu adresu tom tunelu, konfigurirati vrstu tunela, te odabrati odredišnu adresu tj. izlaz iz tunela.

Uputstva za vježbu

1. U ovoj vježbi usmjerivač R2, i sva računala su već konfigurirani. Usmjerivačima R1 i R3 su konfigurirana sučelja G0/0 i G0/1, kao i „*default*“ rute.
2. Kreirajte GRE VPN tunel od LAN 192.168.1.0/24 spojenog na R1 do LAN 192.168.3.0/24 spojenog na R3
3. Konfigurirajte Tunnel0 192.168.2.0/24

   R1-192.168.2.1

   R3-192.168.2.2

4. Koristite statičke rute na R1 i R3 (*next hop address*) kako bi usmjerili promet kroz tunnel0
5. Provjerite funkcionalnost naredbom ping od PC-A do PC-C i obrnuto

CLI naredbe

```
R1(config)#interface tunnel 0 –ulaz u interface mod
R1(config-if)#ip address 192.168.2.1 255.255.255.0 – postavljanje virualne adrese ulaza u tunel
R1(config-if)#tunnel source g0/1
R1(config-if)# tunnel destination 201.150.200.6 – prava adresa 0/1 sučelja na R3
R1(config-if)# tunnel mode gre ip – definira tip tunela
R1(config-if)# exit
R1(config)# ip route 192.168.3.0 255.255.255.0 192.168.2.2

R3(config)# interface tunnel 0 –ulaz u interface mod
R3(config-if)# ip address 192.168.2.2 255.255.255.0 – postavljanje virualne adrese ulaza u tunel
R3(config-if)# tunnel source g0/1
R3(config-if)# tunnel destination 201.150.200.1 – prava adresa 0/1 sučelja na R1
R3(config-if)# tunnel mode gre ip – definira tip tunela
R3(config-if)# exit
R3(config)# ip route 192.168.1.0 255.255.255.0 192.168.2.1
```

8. IPsec tuneliranje

Što je to IPsec?

Internet Protocol Security (**IPsec**) je skup protokola mrežnog sloja koji nam omogućava autentifikaciju i kriptiranje podatkovnih paketa koje šaljemo mrežom. IPsec uključuje protokole za uspostavu uzajamne autentifikacije između spojnih uređaja, te njihovo međusobno dogovaranje o kriptografskim ključevima koji će se koristiti. Ipsec može zaštititi tok podataka između dva uređaja (*host-to-host*), između dva sigurnosna *gateway*-a (*network-to-network*), ili između sigurnosnog *gateway-a* i krajnjeg uređaja (*network-to-host*). *Internet Protocol security* (IPsec) koristi kriptografske sigurnosne usluge koji štite komunikaciju preko IP protokolne mreže. IPsec podržava autetifikaciju krajnje točke na mrežnoj razini, autentifikaciju izvora podataka, nepromjenjivost izvornih podataka, povjerljivost podataka (enkripcija) i zaštitu od presretanja. IPsec skup protokola leži na trećem protokolnom sloju, za razliku od *Transport Layer Security* (TLS) i *Secure Shell* (SSH), koji rade na višim slojevima. Samim time IPsec automatski pruža sigurnost i svim aplikacijama obzirom da pruža sigurnost na trećem sloju.

Najčeće današnje uporabe

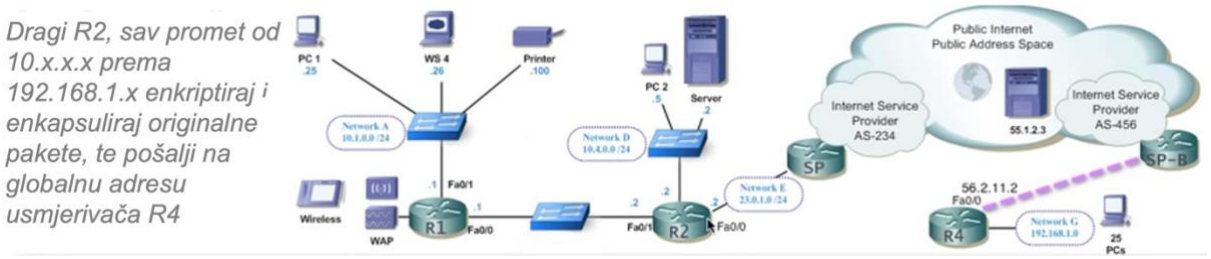
IPsec može se koristiti za uspostavu VPN-ova preko Interneta ili za dodavanje još jednog sloja zaštite na MPLS VPN.

U tunelskom modu, čitav IP paket je kriptiran i autentificiran, te enkapsuliran u novi paket sa novim IP zaglavljem. Tunelski mod se koristi za kreiranje VPN-ova za:

- ✓ *network-to-network* komunikaciju (npr. Između usmjerivača koji povezuju različite mrežne lokacije),
- ✓ *host-to-network* komunikacije (npr. *remote user access*) i
- ✓ *host-to-host* komunikacije (npr. *private chat*).

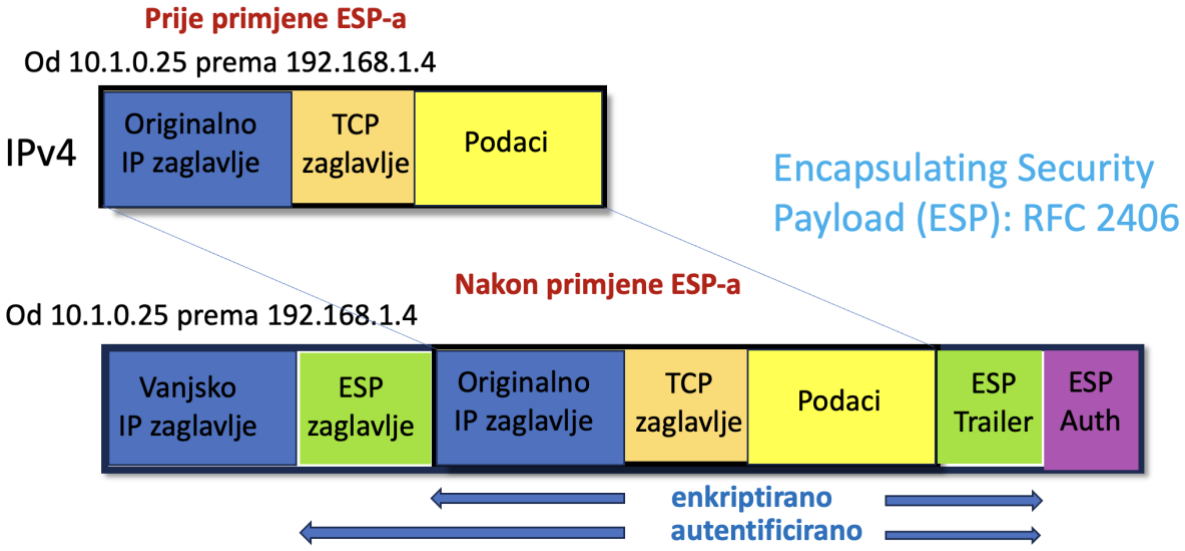
Uporaba Ipsec za VPN povezivanje izdvojenih lokacija na korporativnu mrežu sa zadržavanjem privatnog adresnog plana.

Pri postavljanju IPsec komunikacije u tunelskom modu potrebno je da paketi koji ulaze u tunel izbjegnu klasičnu NAT proceduru (IPsec ne funkcionira uz klasični NAT) već se pri konfiguraciji IPsec tunela odmah definiraju tzv. *peer* globalne adrese (ulazne i izlazne adrese iz tunela).



Slika 8.1. *IPsec tuneliranje*

Mi ovdje želimo reći usmjerivaču R2, da svaki paket koji mu dolazi iz 10.x.x.x mreže, a ima odredište za 192.168.1.x mreži, prvo enkapsulira, kriptira i tek onda pošalje na globalnu adresu usmjerivača R4 tj. na 56.2.11.2



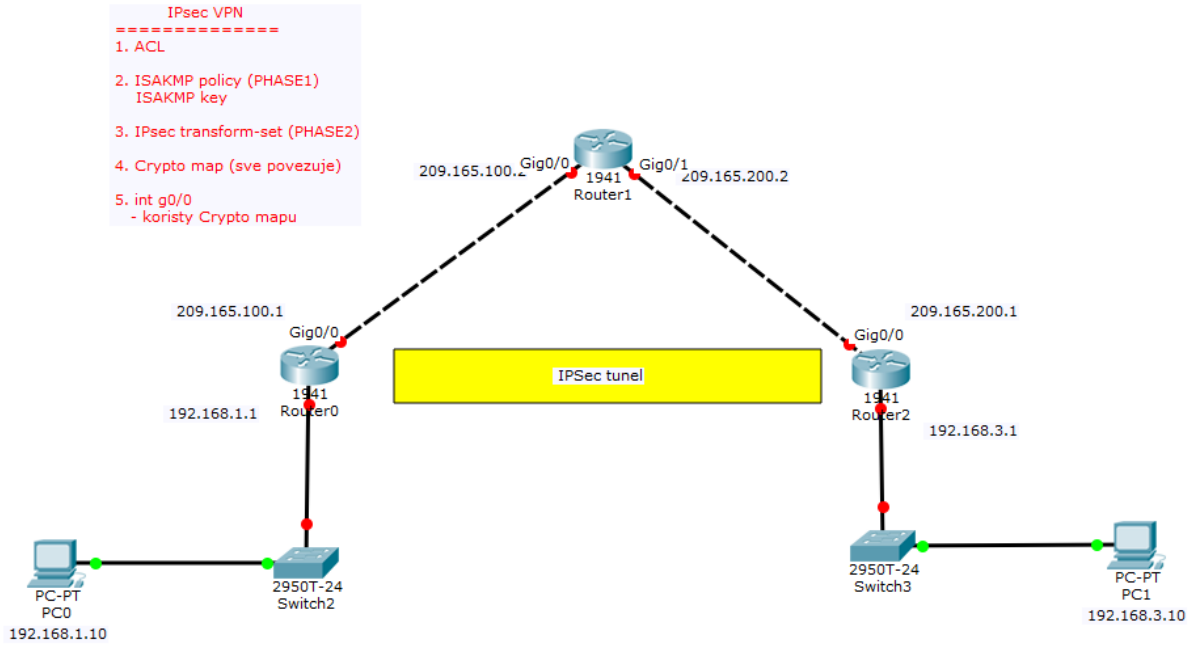
Slika 8.2. *Primjena ESP-a*

Sve što će ISP vidjeti je IP paket koji je odaslan sa 23.0.1.2 na odredišnu adresu 56.2.11.2. Kompletan originalni IP paket je potpuno kriptiran i nevidljiv za ISP-a! Kada taj paket stigne na odredište tj. na usmjerivač R4, on će taj paket dekriptirati i proslijediti na krajnje odredište. Da bismo ostvarili takvu vezu potrebno je kreirati 2 tunela **IKE PHASE1** tunel i **IKE PHASE2** tunel.

IKE phase 1 tunel se koristi za „privatan razgovor“ između usmjerivača R2 i R4 kroz koji oni dogovaraju postavke (npr. dogovor oko tajnog ključa) kojima će izgraditi *IKE phase 2 tunel* odnosno **IPsec tunel** kojim će se transportirati podaci.

Idemo sada u CPT kreirati okruženje u kojemu ćemo pokrenuti IPsec tunel. Nećemo koristiti gornji primjer da bi izbjegli cijelu konfiguraciju NAT-a već ćemo postaviti jednostavnije okruženje kao na slici ispod.

Zadatak – Konfiguracija IPsec tunela



Slika 8.3. *Mrežna topologija za IPsec zadatak [5]*

Sa Router0 kojeg ćemo nazvati R1 i sa Router2 kojeg ćemo nazvati R3 postavljene su statičke rute prema Router1 (kojeg ćemo nazvati ISP), međutim na njemu nećemo postaviti nikakvo usmjeravanje! To znači da paket poslan sa jednog PC-a ne bi trebao stići do drugoga jer ISP usmjerivač ne poznaje put do 192.x.x.x mreža.

1. Početne konfiguracije usmjerivača R1, ISP, i R3.

Router0

```
hostname R1
interface g0/1
ip address 192.168.1.1 255.255.255.0
no shut
interface g0/0
ip address 209.165.100.1 255.255.255.0
no shut
exit
ip route 0.0.0.0 0.0.0.0 209.165.100.2
```

Router1

```
hostname ISP
interface g0/1
ip address 209.165.200.2 255.255.255.0
no shut
interface g0/0
ip address 209.165.100.2 255.255.255.0
no shut
exit
```

Router2

```
hostname R3
interface g0/1
ip address 192.168.3.1 255.255.255.0
no shut
interface g0/0
ip address 209.165.200.1 255.255.255.0
no shut
exit
ip route 0.0.0.0 0.0.0.0 209.165.200.2
```

Naredbom *show ip route* provjeri koje mreže vidi ISP usmjerivač (vidi samo 209.... mreže)

Ping naredbom probaj poslati paket PC-PC

PC-ISP usmjerivač – također ne radi jer ISP nezna put nazad do PC-a

2. Sigurnosne dozvole

Da bi se mogao koristiti IPsec, usmjerivači moraju imati omogućene sigurnosne dozvole!. Provjerite imaju li usmjerivači potrebne dozvole (*security license enabled*). To radimo naredbom

```
R1#show version
```
-nakon nekoliko „entera“ trebali bi vidjeti sliku ispod koja nam kaže da nema dozvole

```
Technology Package License Information for Module:'c1900'

-----------------------------------------------------------------
Technology    Technology-package          Technology-package
              Current       Type          Next reboot
-----------------------------------------------------------------
ipbase        ipbasek9      Permanent     ipbasek9
security      None          None          None
data          None          None          None

Configuration register is 0x2102
 --More--
```

Slika 8.4. *Prikaz stanja sigurnosnih dozvola [5]*

Za aktivaciju **kod ovog tipa usmjerivača**, iz globalnog konfiguracijskog moda koristimo naredbu:

```
R1(config)#license boot module c1900 technology-package securityk9
```

Zatim treba prihvatiti uvjete licence, te snimiti trenutnu kofiguraciju u *startup* konfiguraciju!

```
R1#copy run start
```
, te je sad potrebno restartati uređaj

`R1#reload` - nakon čega će uređaj raditi sa „Evaluacijskom“ dozvolom za korištenje SW za IPsec (inače ga u realnosti treba posebno kupiti)

```
Technology Package License Information for Module:'c1900'

-----------------------------------------------------------------
Technology    Technology-package          Technology-package
              Current       Type          Next reboot
-----------------------------------------------------------------
ipbase        ipbasek9      Permanent     ipbasek9
security      securityk9    Evaluation    securityk9
data          disable       None          None

Configuration register is 0x2102
```

Slika 8.5. *Prikaz stanja sigurnosnih dozvola nakon aktivacije [5]*

Ponoviti sve ovo i za R3 usmjerivač!

3. Konfiguracija IPsec na usmjerivačima na oba kraja tunela (R1 i R3)

R1

Prvo kreiramo *access* listu (nazvanu „100“) iz lijeve u desnu mrežu

```
R1(config)#access-list 100 permit ip 192.168.1.0 0.0.0.255 192.168.3.0 0.0.0.255
```

Ova naredba kreira *access* listu nazvanu „100“ koja samo kreira uvjete za ulaz, ali nema nikave ulogu izrade tunela. *Access* liste možemo koristiti za najrazličitije stvari koje ne moraju imati veze sa IPsec-om. Npr. koristimo ih kod NAT-a, da bi definirali kojim uređajima će biti dozvoljeno da ga koriste. *Access* liste također možemo koristiti za filtriranje ulaznog ili izlaznog prometa,...itd. Znači *access* lista nam samo određuje neko pravilo. Na usmjerivaču možemo istovremeno imati jako puno različitih *access* lista, pa stoga svaku moramo imenovati, kako bi u nekom procesu mogli pozvati baš onu koja nam je potrebna.

U ovom našem slučaju njena funkcija je da kaže usmjerivaču slijedeće: „*Svi paketi koji iz mreže 192.168.1.0 0.0.0.255 idu na odredište u mrežu 192.168.3.0 0.0.0.255 , ući će u naš IPsec tunel*“. Istu listu možemo odmah kreirati i na R3 usmjerivaču, ali sa zamjenjenim mjestima mreža

Idemo sada kreirati „*ISAKMP policy (PHASE1)*“.

ISAKMP - (*Internet Security Asossiation Key Managenet Protocol*)

Ona nam služi za kreiranje inicijalnog tunela, tj. komunikacije dva vanjska sučelja (sa globalnim IP adresama) usmjerivača između kojih se stvara IPsec tunel. Ovdje postavljamo parametre za njihovu komunikaciju prilikom uspostave međusobne autentifikacije. Obzirom da na usmjerivaču možemo imati više IPsec tunela, određene ISAKMP politike (*phase 1*) konfiguriramo kao samostalne objekte. Znači koristi se isti princip kao i za *access* liste. Svaku nazovemo određenim imenom (u našem slučaju nazvati ćemo je „10“), a kasnije ih samo pozivamo.

Sve dole navedene naredbe možemo zajedno kopirati i samo pastirati u R1(config)# .

`crypto isakmp policy 10` - kreiramo *policy* i nazivamo je „10“

```
encryption aes 256              - koristiti ćemo „aes“ enkripciju sa 256 bitnim ključem
authentication pre-share        - za autentifikaciju ćemo koristiti pre-shared ključ
group 5                         - način razmjene pre-shared ključa
!
crypto isakmp key secretkey address 209.165.200.1      - ova naredba postavlja lozinku („secretkey“) i definira adresu ulaznog sučelja usmjerivača na drugoj strani tunela (209.165.200.1). Usmjerivač na drugoj starni mora imati istu konfiguriranu lozinku!!!, ali će imati drugu adresu sučelja druge strane tunela.
```

Idemo sada izvršiti konfiguraciju „*Ipsec transfom-set (Phase2)*“

```
!
crypto ipsec transform-set R1-R3 esp-aes 256 esp-sha-hmac   -prvo kreiramo transformacijski skup pravila i nazivamo ga „R1-R3“, koristiti ćemo „esp-aes“ enkripciju sa „256“bitnim ključem sa „esp-sha-hmac“  tj sa HMAC-variantom SHA autentifikacijskog algoritma pri ESP (Encapsulating Security Payload).
!
```

Sada idemo izraditi našu kripto mapu.

```
crypto map IPSEC-MAP 10 ipsec-isakmp        Nazvati ćemo je IPSEC-MAP 10, a koristiti ćemo je za IPsec i ISAKMP
set peer 209.165.200.1                      Druga strana tunela je 209.165.200.1
set pfs group5                              -pfs (perfect foreward secrecy) biti će grupe 5
set security-association lifetime seconds 86400
set transform-set R1-R3                     poziva transformacijski set R1-R3 koji smo gore kreirali. Obzirom da na nekom usmjerivaču može biti više Ipsec tunela, svakom možemo koristiti drugačiji transformacijski set, također ćemo imati više različitih acccess lista,...itd
match address 100                           Ova naredba kaže da će se ova kripto mapa koristiti za access listu „100“
```

Što nam ova kripto mapa u biti govori? „*Ukoliko promet odgovara access listi 10, tada ga treba poslati na 209.165.200.1, a za kriptiranje podataka koristi transfomacijski set R1-R3“*

Sada idemo primjeniti kriranu kripto mapu na samo sučelje

```
!
interface GigabitEthernet0/0
crypto map IPSEC-MAP                        Ova naredba pokreće ISAKMP proces na ovom sučelju
!
```

Sada sve ponovimo za usmjerivač R3

R3

```
crypto isakmp policy 10
encryption aes 256
authentication pre-share
group 5
!
crypto isakmp key secretkey address 209.165.100.1
!
crypto ipsec transform-set R3-R1 esp-aes 256 esp-sha-hmac
!
crypto map IPSEC-MAP 10 ipsec-isakmp
set peer 209.165.100.1
set pfs group5 set security-association lifetime seconds 86400
set transform-set R3-R1
match address 100
!
interface GigabitEthernet0/0
crypto map IPSEC-MAP
!
access-list 100 permit ip 192.168.3.0 0.0.0.255 192.168.1.0 0.0.0.25
```

Sada naredbom „ping“ provjerite vezu između dva PC-a. Sve bi trebalo raditi.

Napomena 1 - U CPT-u prvih nekoliko pokušaja biti će neuspješno jer mu treba nešto vremena (i neuspješnih pokušaja) da napokon proradi. Što smo sada sa IPsec tunelom dobili? Dobili smo to da ISP nezna niti za jednu privatnu mrežu, a da one međusobno mogu komunicirati i to koristeći njihov privatni adresni plan. To možemo jasno vidjeti iz simulacijskog moda CPT-a.

Napomena 2 – ukoliko se na kolokviju ili ispitu, u nekom zadatku bude tražila uspostava IPsec tunela, ovaj primjer će biti dopušteno koristiti tj. pogledati ga, te prilagoditi. Cilj toga je da studenti ne moraju učiti sve naredbe napamet već trebaju razumjeti proces i trebaju ga znati prilagoditi u datom trenutku na traženi zadatak sa drugačijim mrežama.

9. VLAN (*Virtual* LAN)

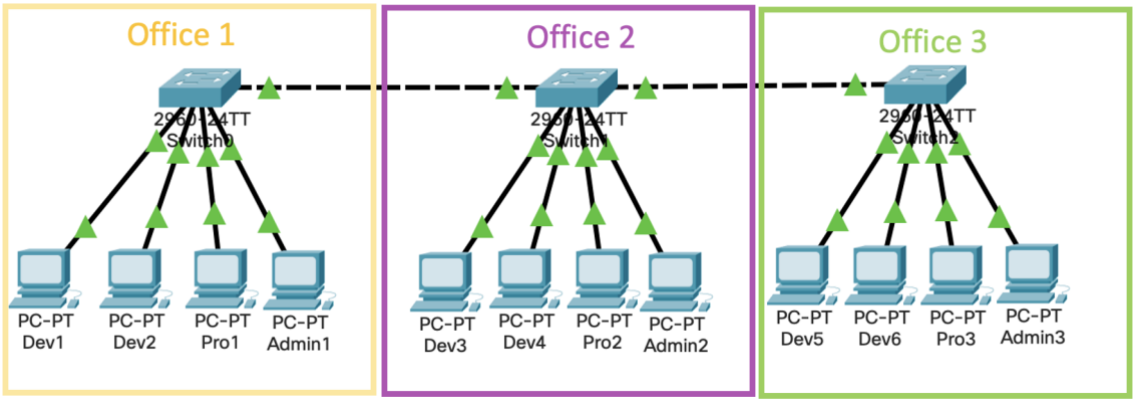
VLAN ili virtualna lokalna mreža predstavlja mrežu u kojoj su uređaji podjeljeni u cjeline ili grupe tj. definira se samostalno prometno područje između određenih uređaja preko kojih mogu razmjenjivati podatke. Promet koji se generira unutar određenog VLAN-a ostaje unutar njega. VLAN je logički konfigurirana mreža, što znači da ne ovisi o fizičkoj topologiji mreže. Dakle, uređaji se mogu nalaziti na različitim lokacijama, u različitim mrežama ali uz VLAN mogu komunicirati međusobno kao da se nalaze u istoj lokalnoj mreži. Upravo zbog ove karakteristike VLAN je dobio na popularnosti. VLAN-ove **konfiguriramo unutar preklopnika**. Administrator mreže mora za svaki priključak (eng. *port*) odrediti pripadnost određenom VLAN-u. VLAN skupine se mogu temeljiti na sličnim potrebama podataka ili sigurnosnim zahtjevima. Na primjer, mogu se izraditi zasebni VLAN-ovi za svaki odjel u firmi kao što su prodaja, računovodstvo i inženjerstvo.

VLAN pruža značajne prednosti:

1) Poboljšana svojstva mreže (ograničen *broadcast*) - Zbog sve većeg broja uređaja na mreži koji se natječu za propusnost dolazi do povećanja količine podataka pa mreža postaje neučinkovita te gubi na raznim svojstvima. Pojavom VLAN-a poboljšavaju se svojstva mreže. Zbog logičke podjele, mreža postaje preglednija, pouzdanija i učinkovitija jer se smanjuje potreba uređaja za obradu podataka koji nije predodređen za njih. Uređaji obrađuju samo one podatke koji dolaze iz istog VLAN-a. Samim time povećava se propusnost jer se *broadcast* promet ne dijeli sa sučeljima izvan određenog VLAN-a.

2) Povećana sigurnost - Znamo da je promet koji se šalje unutar nekog VLAN-a ograničen samo na članove tog VLAN-a. Stoga na neki način možemo VLAN iskoristiti kao mjeru zaštite da se određeni paketi ne dijele sa neovlaštenim korisnicima.

3) **Pojednostavljeno upravljanje mrežom** - Na primjer, ako jedan zaposlenik u određenoj firmi mijenja odjel, dovoljno je samo na preklopniku kroz VLAN konfiguraciju odraditi promjene i on će postati član VLAN-a drugog odjela. Dakle više nema fizičkih pomicanja uređaja, rada sa kabelima i sl. Isti VLAN može se nalaziti na različitim preklopnicima, što znači da ne treba biti na istom fizičkom mjestu da bi se nalazili u istom VLAN-u. Ovime je uvelike pojednostavljeno upravljanje mrežom. Kako bismo bolje razumili VLAN pogledajmo slijedeće primjere:



Slika 9.1. *Mrežna topologija prije VLAN [3]*

- Naša firma ima tri odvojena ureda
- Svi uredi su međusobno spojeni prijenosnom linijom.
- Firma ima 3 odjela: Razvoj, Proizvodnju i Administraciju.
- Razvoj ima 6 računala.
- Proizvodnja ima 3 računala.
- Administracija ima 3 računala.
- Svaki ured ima dva računala iz razvojnog odjela, te po jedno iz proizvodnje i administracije.
- Administracija i proizvodnja imaju osjetljive informacije te moraju biti odvojeni od razvojnog odjela.

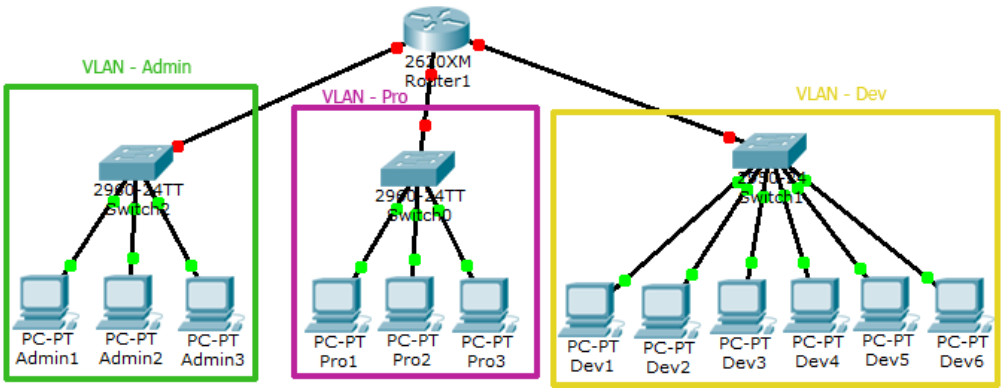
Sa *default* konfiguracijom, sva računala međusobno dijele istu *broadcast* domenu. Razvojni odjel može pristupiti resursima administrativnog i proizvodnog odjela. Računala ne možemo međusobno razdjeljivati subnetiranjem jer se u svakom uredu nalaze izmješana računala svih odjela, a fizički se mogu spojiti samo na preklopnik u tom uredu. Za podjelu subnetiranjem morali bi u svaki ured postaviti onoliko preklopnika koliko ima odjela,...što bi bilo vrlo skupo i nezgrapno rješenje.

Sa VLAN-om možemo kreirati logičke granice unutar jedne fizičke mreže.

Pretpostavimo da smo kreirali tri VLAN-a unutar naše fizičke mreže i tim VLAN-ovima smo dodijelili pripadajuća računala.

- VLAN **Admin** za Administrativni odjel
- VLAN **Dev** za Razvojni (*Development*) odjel
- VLAN **Pro** za Proizvodni odjel

Nismo napravili nikakve **fizičke** promjene na mreži (prespajali kablove, dodavali uređaje, itd.), već smo samo uporabom određenih SW naredbi na preklopniku (*switch*) kreirali **logičku podjelu** i grupirali uređaje po pripadnosti odjelima. Ovakve grupe (VLAN-ovi) više ne mogu komunicirati jedan sa drugim. Da bi im se to ipak omogućilo moraju se svi spojiti na usmjerivač. Naša mreža sada **logički** izgleda kao na slici ispod.



Slika 9.2. *Logička mrežna topologija nakon VLAN[3]*

Pomoću VLAN-a razdvojili smo jednu veliku mrežu u tri manje. Ove mreže međusobno ne dijele *broadcast* što popravlja mrežne performanse. VLAN također podiže razinu sigurnosti. Razvojni odjel sada ne može direktno pristupati resursima Proizvodnog i Administrativnog odjela za koje smo rekli da posjeduju osjetljive informacije. Različiti VLAN-ovi mogu međusobno komunicirati samo preko usmjerivača (*router*) na kojem pak možemo primjeniti najrazličitije sigurnosne opcije.

Metode dodjele članstva u VLAN-u

Postoje dvije metode da se nekom uređaju dodjeli članstvo u VLAN-u.

1. **Statička metoda i**
2. **Dinamička metoda**

Ove metode određuju preklopniku (*switch*) način na koji će svojim sučeljima dodjeljivati članstvo u različitim VLAN-ovima.

Statička metoda

Statičko dodjeljivanje članstva najčešća je i najsigurnija metoda. Vrlo je jednostavna za postavljanje i održavanje. Kod ove metode mi manualno dodjeljujemo članstvo određenom sučelju (*port*) preklopnika (switch) nekom VLAN-u. VLAN-ovi kreirani na ovaj način najčešće se nazivaju „**port-based VLAN“** jer je članstvo nekog uređaja u određenom VLAN-u definirano sučeljem na preklopniku putem kojeg je uređaj spojen na mrežu. Bilo koje računalo koje se priključi na mrežu putem tog sučelja, automatski postaje član VLAN-a. Ova metoda je i najsigurnija jer će svaki preklopnik zadržati iste postavke sve dok ga mi opet manualno ne re-konfiguriramo. Ova metoda dobra je za situacije gdje kretanje korisnika unutar mreže želimo držati pod nadzorom, tj. gdje je potreba za mobilnošću djelatnika zanemariva.

Dinamička metoda

Kod dinamičke metode, članstvo nekog sučelja (*port*) preklopnika u određenom VLAN-u dodjeljuje se automatski ovisno o uređaju koji je trenutno spojen na to sučelje. Ukoliko želimo primjeniti ovakvu konfiguraciju, moramo jedan preklopnik u mreži konfigurirati kao „server“. Taj server u svojoj bazi podataka ima pohranjene specifične informacije uređaja (npr. MAC adresa ili IP adresa) kojima je dozvoljeno da postanu članovi određenog VLAN-a. Preklopnik (*switch*) koji radi u „server“ modu naziva se VMPS (*VLAN Membership Policy Server*). Samo preklopnici novijih generacija mogu se konfigurirati kao VMPS. Preklopnici starijih generacija i manjih mogućnosti rade u klijentskom modu, te povlače potrebne VLAN informacije iz VMPS preklopnika. Dinamički VLAN podržava „*plug and play*“ mobilnost. Pogodan je za firme gdje je potrebna velika mobilnost djelatnika. Npr. imamo firmu koja ima svoje poslovnice na više odvojenih lokacija (npr. gradova). Djelatniku firme je dodjeljeno prijenosno računalo na kojem je statički postavljena IP adresa (ili je pak poznata MAC adresa računala), a ta informacija pohranjena je u VMPS serveru te definira članstvo tog računala (a samim time i djelatnika) određenom VLAN-u. Djelatnik sa tim računalom može otići u bilo koji ured te firme, te se spojiti na bilo koji mrežni priključak i taj **priključak** (tj. sučelje ili *port*) će odmah biti dodjeljen određenom VLAN-u, a samim time djelatnik će imati pristup svim svojim mrežnim resursima. Kod statičke metode, mrežni administrator bi svaki put morao manualno rekonfigurirati članstvo porta, što bi u realnom životu znatno usporavalo poslovanje i tražilo

veliki broj mrežnih administratora. Međutim, kod dinamičke metode spušta se razina sigurnosti. Ukoliko je nekome neovlaštenome poznata IP adresa (ili MAC broj) prijenosnog računala sa početka ove priče, može tu IP adresu postaviti na neko drugo računalo te neovlašteno pristupiti u članstvo VLAN-a. (softverski se može lažirati i MAC adresa računala)

U ovom kolegiju dalje ćemo obrađivati samo Statičku metodu kreiranja VLAN-a.

VLAN vrste veza

Preklopnici podržavaju dvije vrste veza za VLAN povezivanja

1. *Access (*pristupni) *link*
2. *Trunk link*

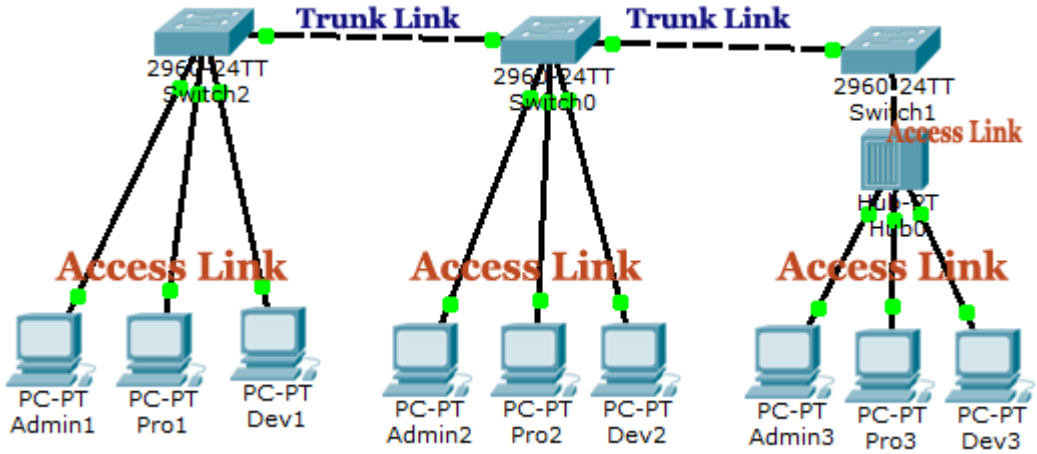
***Access link* (pristupna veza)**

Access link je vrsta veze kod koje se sučelje preklopnika spaja sa nekim uređajem koji ima standardizirano Ethernet NIC sučelje koje samo „razumije“ IEEE 802.3 ili Ethernet II okvire (*frames*). Ili da pojednostavnimo,.. *access link* je veza kojom povezujemo uređaje na preklopnik. *Access link* je uvijek dodjeljen samo jednom VLAN-u.

Trunk link

Trunk link je veza kojom spajamo preklopnik sa uređajima koji su u stanju razumjeti višestruke VLAN-ove. U realnom životu, *trunk link* su veze kojima međusobno povezujemo preklopnik sa drugim preklopnikom ili pak preklopnik sa usmjerivačem.

Kod *trunk* veza originalni Ethernet okvir je modificiran kako bi mogao prenositi VLAN informacije.



Slika 9.3. *Access i trunk veze [3]*

***Trunk* označavanje (*tagging*)**

Kod tzv. *trunk* veza, mi u jednom fizičkom kanalu (jedan spojni kabel) kreiramo više logičkih veza i to po jednu za svaki VLAN koji smo kreirali jer tom vezom (istim kabelom) prolazi komunikacija svih VLAN-ova. Kako bi preklopnici znali koji paket pripada kojem VLAN-u, oni moraju uvesti **označavanje(*tagging*)** svakog paketa kojeg šalju kroz *trunk link*. Preklopnik koji prima paket sa *trunk* veze pomoću *VLAN identifier* oznake dodane Ethernet okviru, zna koji VLAN je poslao taj paket, te na koja svoja sučelja ga smije prosljediti (samo na ona sučelja tj. *access linkove* koji pripadaju baš tom VLAN-u).

Ukoliko ovakav modificirani Ethernet okvir stigne na standardno sučelje nekog uređaja (npr. mrežna kartica na PC-u), ono ga neće razumjeti i paket će biti odbačen.

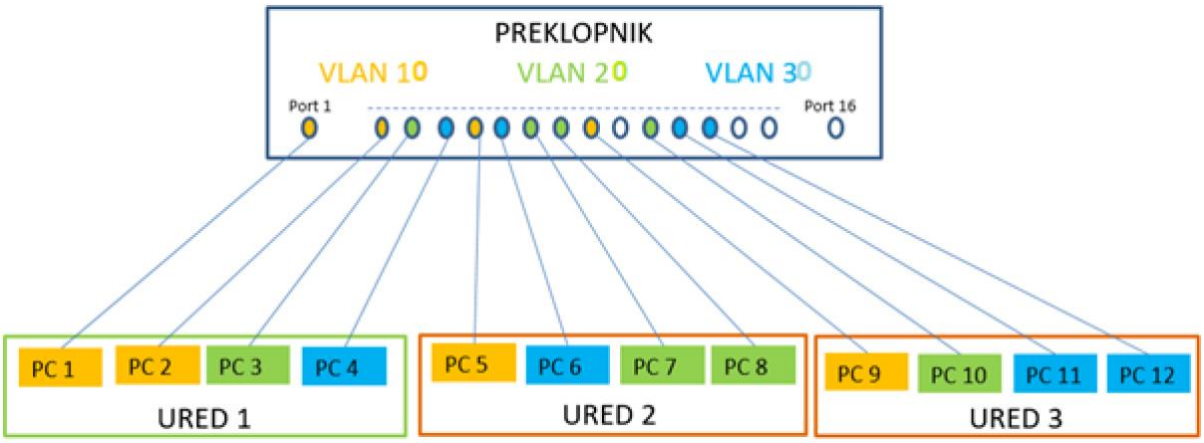
Preklopnici uglavnom podržavaju dvije vrste *Ethernet trunking* ozačavanja paketa:

1. ISL [Inter Switch Link, Cisco's proprietary protocol for Ethernet] – samo CISCO preklopnici
2. Dot1q [IEEE's 802.1Q, protocol for Ethernet] – otvoreni standard, svi proizvođači

Idemo sada kroz nekoliko primjera razjasniti kako možemo kreirati VLAN-ove sa *access* i sa *trunk* vezama, te vidjeti razloge zašto imamo dvije vrste veza u VLAN tehnologiji.

Kreiranje VLAN-a uporabom isključivo *access* veza

Primjer 1 – Najjednostavnija oblik VLAN-a gdje imamo npr. samo preklopnik sa 16 sučelja (*port*) i 12 računala koja se nalaze u 3 ureda koja se nalaze relativno blizu te se sva računala Ethernet kabelima mogu direktno priključiti na preklopnik.



Slika 9.4. *Jednostavni VLAN samo sa access vezama*

U ovom slučaju koristimo samo *Access link* veze!

Sučelja na koja su spojena računala 1,2,5 i 9 definiramo da pripadaju VLAN-u 10

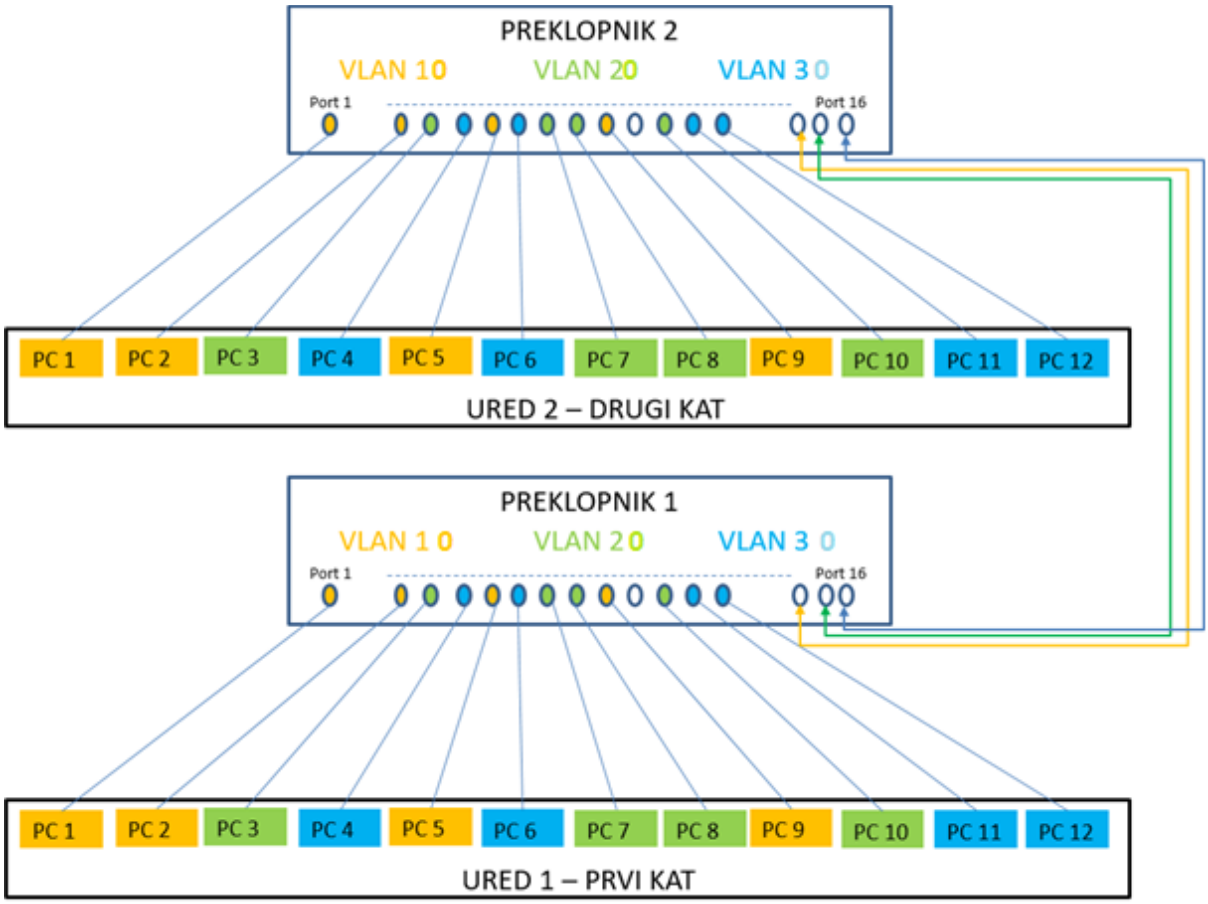
Sučelja na koja su spojena računala 3,7,8 i 10 definiramo da pripadaju VLAN-u 20

Sučelja na koja su spojena računala 4,6,11 i 12 definiramo da pripadaju VLAN-u 30

Sučelja 9, 14, 15 i 16 ostaju slobodna

Sve VLAN-ove kreiramo na tom preklopniku i VLAN komunikacija ostaje unutar preklopnika.

Primjer 2 – Imamo 2 velika ureda, po jedan na svakom katu zgrade. U svakom uredu imamo 12 računala koja su podjeljena u 3 VLAN-a. Na svakom katu imamo jedan preklopnik na koji su spojena računala.



Slika 9.5. *Vlan sa dva preklopnika i samo access vezama*

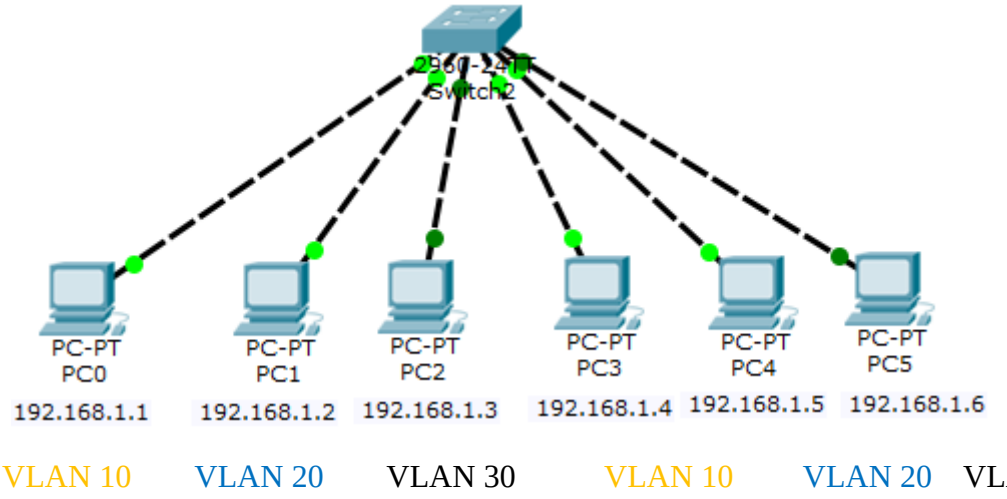
Računala su na oba preklopnika po sučeljima spojena jednako kao i u prvom primjeru, ali smo ovdje još definirali sučelja 14, 15 i 16 na oba preklopnika. Pošto je ovo VLAN koji se temelji na definiranom sučelju tj. ne ovisi o nikakvim drugim kriterijima, za svaki VLAN potrebno nam je jedno sučelje koje će služiti za povezivanje istih VLAN-ova na dva preklopnika

- **Sučelja 14 na oba preklopnika definirali smo kao članove VLAN 10**- Na ovaj način smo omogućili računalima iz VLAN 1 koji se nalaze u Uredu1 da komuniciraju sa računalima koji pripadaju VLAN 1 iz Ureda 2
- **Sučelja 15 na oba preklopnika definirali smo kao članove VLAN 20** - omogućili računalima iz VLAN 2 koji se nalaze u Uredu1 da komuniciraju sa računalima u VLAN 2 iz Ureda 2
- **Sučelja 16 na oba preklopnika definirali smo kao članove VLAN 30** - omogućili računalima iz VLAN 3 koji se nalaze u Uredu1 da komuniciraju sa računalima u VLAN 30 iz Ureda2

I dalje koristimo samo *Access* veze što nam olakšava konfiguraciju, ali vidimo da sada koristimo još dodatna 3 sučelja. **Potreban nam je onoliki broj dodatnih sučelja koliko imamo VLAN-ova**.

Zamislite sada da imate 10 VLAN-ova i urede na 10 katova što je čest slučaj u većim firmama. Na svakom preklopniku po 10 sučelja bilo bi izgubljeno samo za međusobno spajanje preklopnika. Trebalo bi provlačiti i 10 dodatnih kabela između katova. Sučelja i kabeli koštaju, a ponekad nema ni mjesta za provući te kablove između katova. **Iz tih razloga, ukoliko imamo veće mreže, neophodno je uvođenje *trunk* veza!**

Zadatak 1. – U mrežnoj topologiji kao na slici ispod na preklopniku kreirajte 3 VLAN sa po dva računala u svakom.



Slika 9.6. *Mrežna topologija za VLAN zadatak 1*

Prvo je potrebno računalima dodati IP adrese, a zatim ih spojiti redom na FastEthernet sučelja preklopnika (od FastEthernet 0/1 do FastEthernet 0/6).

Nakon toga krećemo u konfiguriranje VLAN-ova na preklopniku. Kod kreiranja VLAN-ova na preklopniku bitno je razlikovati dva pojma ***VLAN name*** i ***VLAN index (ili broj)***.

- ***VLAN name*** služi isključivo nama kako bi lakše razlikovali VLAN-ove, pa tako nekom VLAN-u možemo dodijeliti ime „proizvodnja“ ili „administracija“, itd.
- ***VLAN index (ili broj)*** pak je broj pomoću kojega preklopnici razlikuju VLAN-ove i sve njihove operacije vezane za VLAN odvijaju se temeljem tih brojeva.

```
Switch>enable
Switch#configure terminal
Switch(config)#vlan 10   - kreiranje VLAN 10 („10“-predstavlja index VLAN)
Switch(config-vlan)#exit
Switch(config)#vlan 20   - kreiranje VLAN 20
Switch(config-vlan)#exit
Switch(config)#vlan 30   - kreiranje VLAN 30
Switch(config-vlan)#exit
Switch(config)#interface fastEthernet 0/1
Switch(config-if)#switchport access vlan 10  -definira da je ovo sučelje access port i da pripada VLAN 10
```

```
Switch(config-if)#exit
Switch(config)#interface fastEthernet 0/2
Switch(config-if)#switchport access vlan 20      -definiran da je ovo sučelje access port i
da pripada VLAN 20
Switch(config-if)#exit
Switch(config)#interface fastEthernet 0/3
Switch(config-if)#switchport access vlan 30  -definira da je ovo sučelje access port i da
pripada VLAN 30
Switch(config-if)#exit
Switch(config)#interface fastEthernet 0/4
Switch(config-if)#switchport access vlan 10
Switch(config-if)#exit
Switch(config)#interface fastEthernet 0/5
Switch(config-if)#switchport access vlan 20
Switch(config-if)#exit
Switch(config)#interface fastEthernet 0/6
Switch(config-if)#switchport access vlan 30
Switch(config-if)#exit
Switch(config)#
Switch#
```

Sada testiramo našu konfiguraciju. Računala koja su unutar istog VLAN-a trebala bi moći komunicirati, a komunikacija ne bi trebala biti moguća sa ostalim računalima. Komunikaciju testiramo naredbom *ping*.

NAPOMENE:

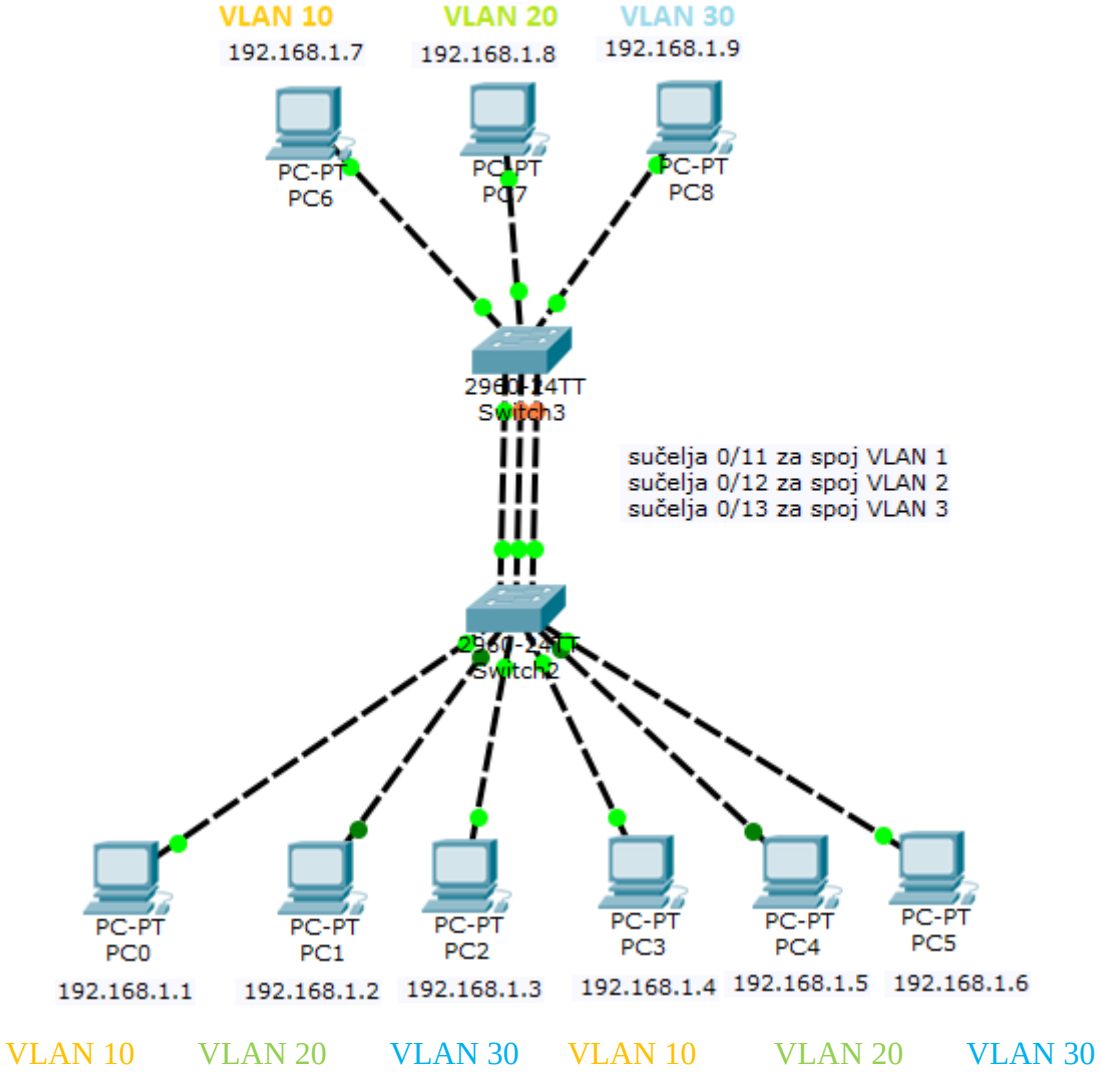
1. Konfiguracija VLAN-ova na preklopniku može se izvršiti i preko grafičkog sučelja. Kada otvorimo grafičko sučelje vidimo da je tamo uvijek kreiran VLAN 1 (kao i još neki sistemski) , te da ga ne možemo izbrisati. Obično se koristi za održavanje
2. Obzirom da u zadatku mi nismo definiral VLAN „ime“ – preklopnik je ime generirao automatski npr. za VLAN 30, dodjelio je ime „VLAN0030“
3. Da smo iz konfiguracijskog moda željeli odmah imenovati VLAN to bi radili naredbom ***name*** nakon kreiranja VLAN indeksa. Vidi primjer ispod.

   ```
   Switch(config)#vlan 30
   Switch(config-vlan)#name proizvodnja
   Switch(config-vlan)#exit
   ```

4. Sva sučelja su u većini preklopnika po standardu su-postavljena u *access* mod pa za naš zadatak nismo trebali ništa mjenjati. Mod rada sučelja (*access* ili *trunk*) mjenja se naredbom

   ```
   Switch (config)#interface FastEthernet0/8
   Switch (config-if)#
   Switch (config-if)#switchport mode trunk
   Switch (config-if)#
   ```

Zadatak 2. – U mrežnu topologiju iz zadatka 1 dodajte još tri računala i još jedan preklopnik kao na slici ispod, te na drugom preklopniku kreirajte 3 VLAN sa po jednim računala u svakom. Izvršite spajanje ta dva preklopnika kao bi sva računala istih VLAN-ova mogla komunicirati.



Slika 9.7. *Mrežna topologija za VLAN zadatak 2*

Prvo dodjeliti IP adrese računalima, spojiti ih na preklopnik, te spojiti preklopnike međusobno sa tri veze (po jedna za svaki VLAN). Sada je potrebno kreirati VLAN-ove na drugom preklopniku, definirati sva sučelja na drugo preklopniku, a zatim kreirati dodatna sučelja i na prvom preklopniku.

Konfiguracija drugog preklopnik (na slici označen kao Switch3)

```
Switch>enable
Switch#configure terminal
Switch(config)#vlan 10   - kreiranje VLAN 10
Switch(config-vlan)#exit
Switch(config)#vlan 20   - kreiranje VLAN 20
Switch(config-vlan)#exit
Switch(config)#vlan 30   - kreiranje VLAN 10
Switch(config-vlan)#exit
Switch(config)#interface fastethernet 0/1
```

```
Switch(config-if)#switchport access vlan 10      -definiranje da je ovo sučelje access port
i da pripada VLAN 10
Switch(config-if)#exit
Switch(config)#interface fastethernet 0/2
Switch(config-if)#switchport access vlan 20      -definiranje da je ovo sučelje access porti
da pripada VLAN 20
Switch(config-if)#exit
Switch(config)#interface fastethernet 0/3
Switch(config-if)#switchport access vlan 30      -definiranje da je ovo sučelje access porti
da pripada VLAN 30
Switch(config-if)#exit
Switch(config)#interface fastethernet 0/11
Switch(config-if)#switchport access vlan 10      -definiranje da je ovo sučelje access porti
da pripada VLAN 10
Switch(config-if)#exit
Switch(config)#interface fastethernet 0/12
Switch(config-if)#switchport access vlan 20      -definiranje da je ovo sučelje access porti
da pripada VLAN 20
Switch(config-if)#exit
Switch(config)#interface fastethernet 0/13
Switch(config-if)#switchport access vlan 30      -definiranje da je ovo sučelje access porti
da pripada VLAN 30
Switch(config-if)#exit
Switch(config)#
```

Dodatna konfiguracija prvog preklopnika (na slici označen kao Switch2)

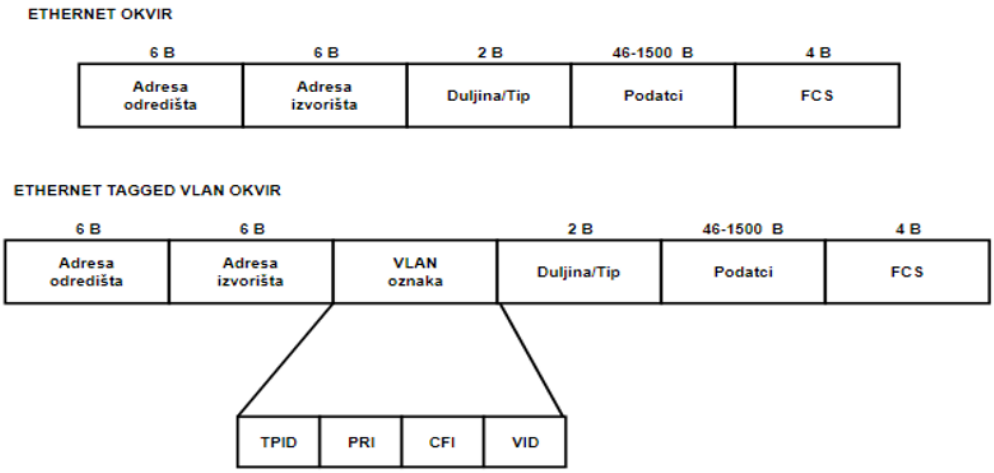
```
Switch>enable
Switch#configure terminal
Switch(config)#interface fastethernet 0/11
Switch(config-if)#switchport access vlan 10
Switch(config-if)#exit
Switch(config)#switchport access vlan 10
Switch(config-if)#interface fastethernet 0/12
Switch(config-if)#switchport access vlan 20
Switch(config-if)#exit
Switch(config)#interface fastethernet 0/13
Switch(config-if)#switchport access vlan 30
Switch(config-if)#exit
Switch(config)#
```

Kreiranje tzv. „označenog“ (*tagged*) VLAN-a

Kao što smo već prije spomenuli, kod **statičke medode** kreiranja VLAN-a (tzv. ***port-based VLAN***), pripadnost nekom VLAN-u određena je priključnim sučeljem (*port*) na preklopniku, a mi smo ti koji definiramo koje sučelje (port) preklopnika će pripadati kojem VLAN-u. Također smo spomenuli da postoje dvije vrste veza:

Access link– koji radi po čistom Ethernet standardu (okviri IP paketa se ne mjenjaju)

Trunk link– služi za međusobno povezivanje preklopnika ili za veze preklopnika s usmjerivačem. Obzirom da ovdje kroz jednu te istu fizičku vezu prolazi promet različitih VLAN-ova, preklopnik koji šalje paket na trunk link mora drugačije **označiti** (*tagging*) okvire IP paketa svakog pojedinog VLAN-a kako bi preklopnik koji prima pakete sa *trunk link-a* znao kojem VLAN-u pripada koji paket.



Slika 9.8. *Razlika između Standardnog i „označenog“ Ethernet okvira*

- TPID (eng. Tag Protocol Identifier) – služi kao identifikator okvira 802.1Q. Kod neoznačenih okvira nalazi se na polju Legnth/Type.
- PRI (eng. Priority) – polje prioriteta, definira 8 razina, 0 kao najmanji i 7 kao najveći.
- CFI (eng. Canonical Format Indicator) – koristi se zbog kompatibilnosti Ethernet i TokenRing mreža. U Ethernet mrežama CFI je postavljen na 1, a u Token Ring mrežama na 0.
- VID (eng. VLAN Identifier) – polje koje određuje kojem VLAN-u pripada okvir.

Kako bi preklopnici mogli „označavati“ Ethernet okvire morao se uvesti protokol koji će to raditi, a sva sučelja koja su uključena u trunk link komunikaciju moraju biti u stanju „čitati“ te proširene okvire. Dva protokola za označavanje okvira danas imaju najveći značaj na tržištu, a to su:

1. Dot1q [*IEEE's 802.1Q, protocol for Ethernet*] – otvoreni standard, svi proizvođači
2. ISL [*Inter Switch Link, Cisco's proprietary protocol for Ethernet*] – samo CISCO preklopnici

... ali čak i Cisco u novijim verzijama uređaja u potpunosti prelazi na IEEE 802.1Q

Konfiguriranje „označenog“ VLAN-a

Označeni (*tagged*) VLAN definiran je slijedećim karakteristikama:

- VLAN indeksom,
- Imenom VLAN-a,
- Označenim i neoznačenim sučeljima tj. *access* i *trunk* sučeljima,
- Identifikatorom sučelja VLAN-a (*Port VLAN Identifier* (PVID)),
- Općim pravilima za stvaranje označenog VLAN-a.

VLAN Indeks

Svaki VLAN u mreži mora imati jedinstveni broj. Ovaj broj se naziva VLAN ID te jedinstveno identificira VLAN u preklopniku i u mreži (svim povezanim preklopnicima).

VLAN Ime

VLAN-u možemo dati i jedinstveno ime. To ime može biti vezano uz funkciju mrežnih uređaja koji su članovi VLAN-a, npr. kao što su „prodaja“, „proizvodnja“ ili „inženjering“. Ukoliko mi ne definiramo VLAN ime, to će automatski napraviti sam preklopnik.

Označena i neoznačena sučelja tj. *trunk* ili *access*

Kada smo odredili da je sučelje član označenog VLAN-a, moramo još odrediti da li je označeno (*trunk*) ili neoznačeno (*access*). VLAN može sadržavati označena i neoznačena sučelja. Prijenos paketa iz označenog sučelja razlikuje se od prijenosa paketa putem neoznačenog sučelja.

- Kad preklopnik šalje paket sa označenog sučelja, informacija o VLAN pripadnosti unutar Ethernet okvira se zadržava pri prijenosu okvira na sljedeći mrežni uređaj.
- Ako pak preklopnik paket šalje sa neoznačenog sučelja, VLAN oznaka je uklonjena iz okvira prije nego što se prenese na drugi mrežni uređaj.

IEEE **802.1Q** standard opisuje kako se koristi VLAN označavanje unutar paketa da bi se proslijedio ili odbacio promet kroz preklopnik. Ako dolazni paket ima VLAN oznaku koja odgovara jednoj od skupina ID-ova čiji je sučelje član, paket se prihvaća i prosljeđuje na odgovarajuće sučelje unutar tog VLAN-a. Ako se oznaka VLAN-a dolaznog paketa ne poklapa s nekim od skupina ID-ova koji su dodijeljeni sučeljima, paket se odbacuje.

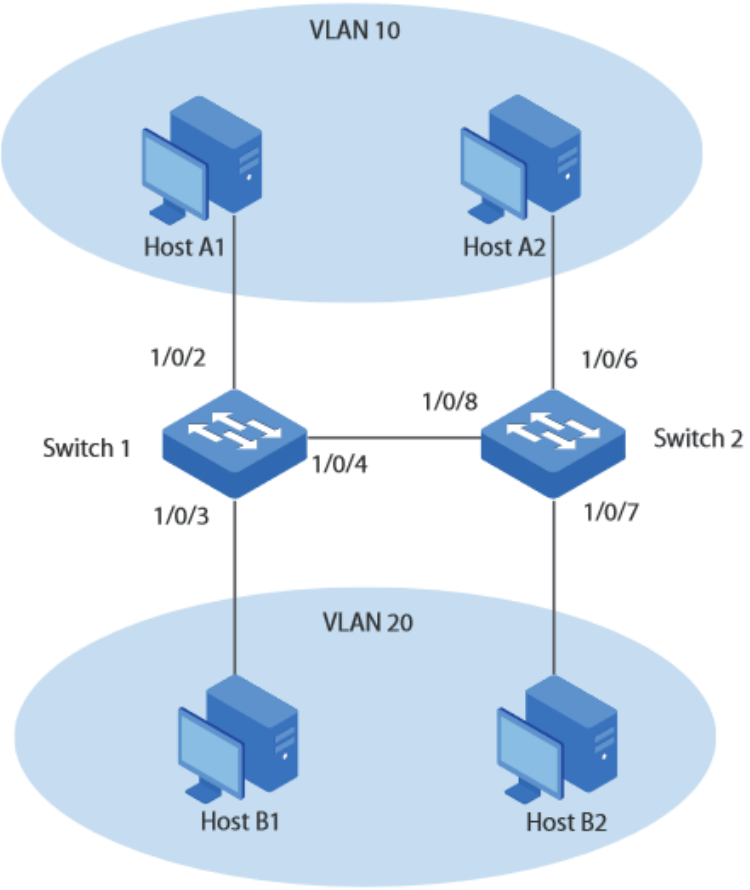
Označeno (*trunk*) sučelje može biti član više VLAN-ova. (npr. sučelja kroz koja međusobno povezujemo VLAN-ove)

Identifikator sučelja (*port-a*) VLAN-a (PVID)

Svakom sučelju prilikom njegove konfiguracije moramo dodjeliti PVID tj. broj VLAN-a kojem ono pripada. Kada je neoznačeni paket (npr. kojeg šalje računalo) primljen na sučelju u označenom (*tagged*) VLAN-u, automatski je dodijeljen VLAN-u kojemu to sučelje pripada. Odlučujući faktor u tom procesu je Identifikator sučelja VLAN-a (PVID). Sva sučelja preklopnika, označena (*tagged*) i neoznačena (*access*), moraju imati dodijeljen PVID. Zadana početna vrijednost (tvornički postavljena) PVID-a za svako sučelje je „1“ jer čak ako mi nismo

na preklopniku kreirali VLAN-ove, tvornički je već kreiran VLAN 1 u koji su uključena sva sučelja (pa nama izgleda kao i da ne postoji jer ništa ne dijeli). Preklopnik povezuje primljeni neoznačeni paket sa VLAN ID-om čiji se broj poklapa sa PVID-om koji je dodijeljen sučelju, a paket se prosljeđuje samo onim sučeljima koji su članovi.

Primjer – Konfiguriraj „označeni“ (*tagged*) VLAN kao na slici preko naredbenog moda (CLI)



Slika 9.9. *Mrežna topologija za primjer VLAN konfiguracije – zadatak 1*

- Računala Host A1 i Host A2 su iz Odjela A, dok su Host B1 i Host B2 iz Odjela B.
- Preklopnici Switch 1 i Switch 2 smješteni su na dvije različite lokacije.
- Host A1 i Host B1 spojeni su na sučelja 1/0/2 i 1/0/3 preklopnika Switch 1, a Host A2 i Host B2 spojeni su na sučelja 1/0/6 i 1/0/7 preklopnika Switch 2.
- Sučelje 1/0/4 preklopnika Switch 1 spojeno je sa sučeljem 1/0/8 preklopnika Switch 2.

U nastavku će po koracima biti objašnjena konfiguracija preklopnika Switch 1.

Switch 2 konfigurira se istom analogijom (koraci su isti, mjenjaju se oznake korištenih sučelja)

Konfiguracija iz CLI (*Command Line Interface*)

- Kreiraj VLAN 10 za Department A, i konfiguriraj ime Department-A.
- Kreiraj VLAN 20 za Department B, i konfiguriraj ime Department-B.

```
Switch_1#configure
Switch_1(config)#vlan 10
Switch_1(config-vlan)#name Department-A
Switch_1(config-vlan)#exit
Switch_1(config)#vlan 20
Switch_1(config-vlan)#name Department-B
Switch_1(config-vlan)#exit
```

Postavi mod sučelja 1/0/2 i 1/0/3 na *Access*, a onda pridruži 1/0/2 VLAN-u 10, te sučelje 1/0/3 VLAN-u 20.

```
Switch_1(config)#interface gigabitEthernet 1/0/2
Switch_1(config-if)#switchport mode access
Switch_1(config-if)#switchport access vlan 10
Switch_1(config-if)#exit
Switch_1(config)#interface gigabitEthernet 1/0/3
Switch_1(config-if)#switchport mode access
Switch_1(config-if)#switchport access vlan 20
Switch_1(config-if)#exit
```

Postavi mod sučelja 1/0/4 na *Trunk*, a onda mu dodjeli oba VLAN-a (VLAN 10 i VLAN 20).

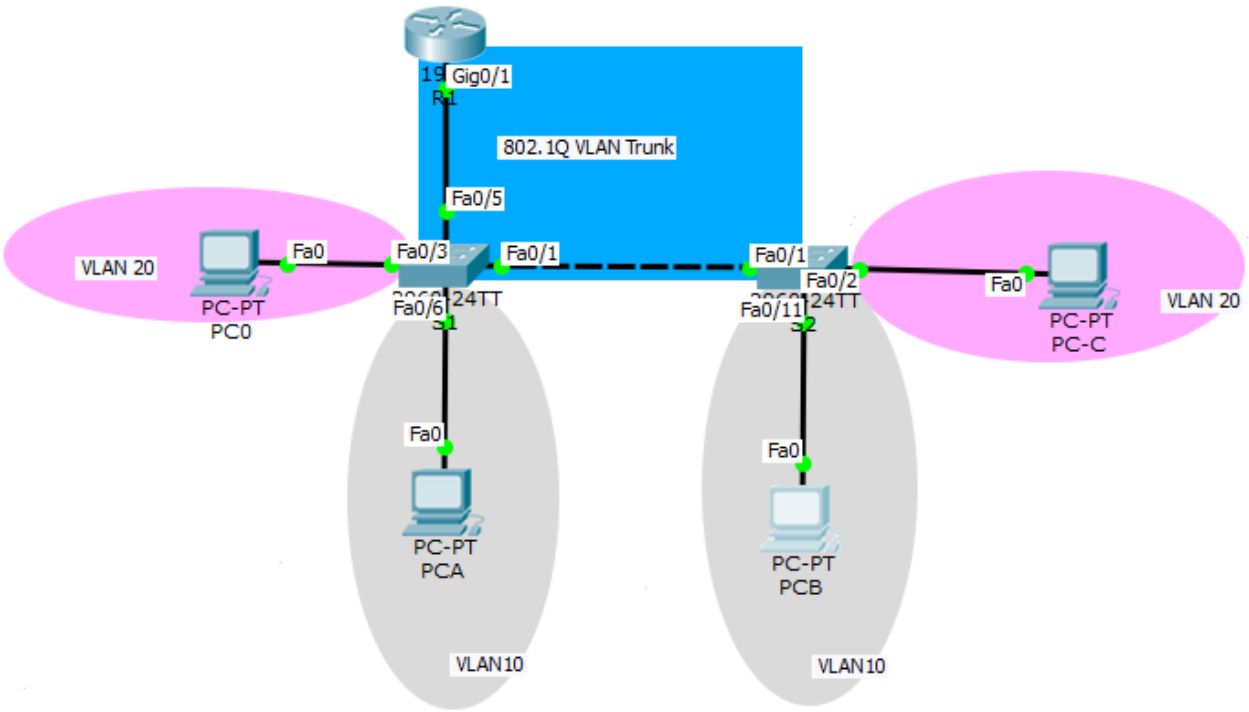
```
Switch_1(config)#interface gigabitEthernet 1/0/4
Switch_1(config-if)#switchport mode trunk
Switch_1(config-if)#switchport trunk allowed vlan 10,20
Switch_1(config-if)#end
Switch_1#copy running-config startup-config
```

Zadatak – Uspostava inter VLAN komunikacije „dot1Q“ metodom

U zadanoj mrežnoj topologiji kao na slici ispod konfigurirati osnovne postavke uređaja, kreirati VLAN-ove, dodjeliti PVID svim sučeljima te konfigurirati 802.1Q *trunk* između preklopnika. Lijevi preklopnik nazvati S1, a desni S2. Kada smo to napravili, uređaji iz jednog VLAN-a više neće moći komunicirati s uređajima koji pripadaju drugom VLAN-u.

Nakon toga ponovno omogućiti komunikaciju između VLAN-ova ali sada preko usmjerivača.

Kako bi to bilo moguće, svaki VLAN mora biti unutar različite podmreže jer usmjerivači usmjeravaju promet između različitih MREŽA!



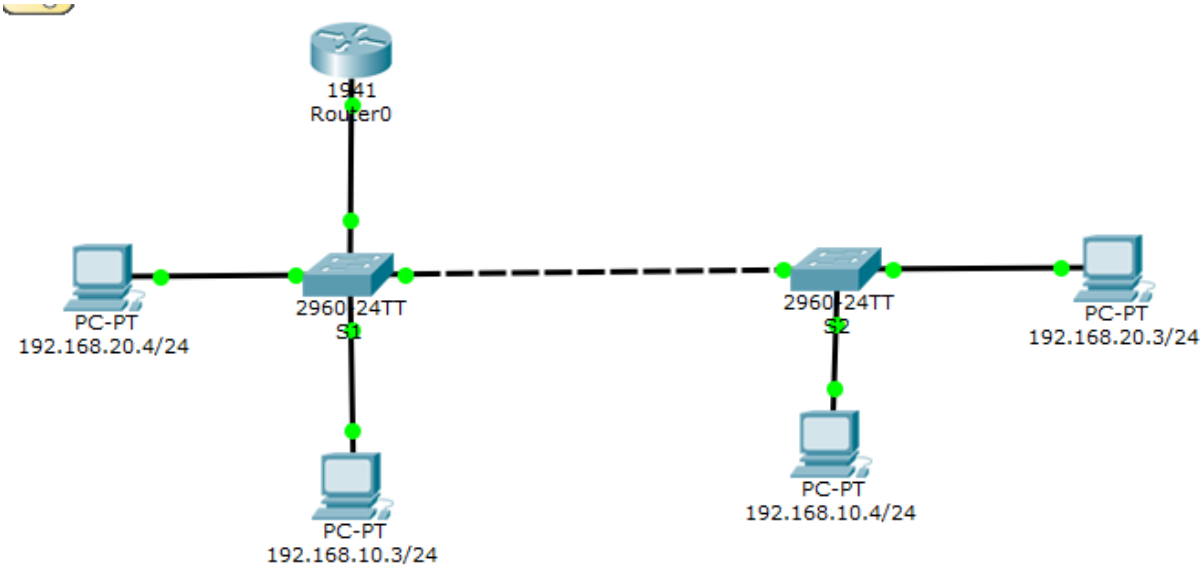
Slika 9.10. *Mrežna topologija za primjer VLAN konfiguracije – zadatak 2*

Tablica adresa:

Device	Interface	IP Address	Subnet Mask	Default Gateway
PC-A	NIC	192.168.10.3	255.255.255.0	192.168.10.1
PC-B	NIC	192.168.10.4	255.255.255.0	192.168.10.1
PC-C	NIC	192.168.20.3	255.255.255.0	192.168.20.1

Tablica 9.1. Tablica adresa za VLAN zadatak 2

PC0 ima adresu 192.168.20.4, masku podmreže 255.255.255.0 i *default gateway* 192.168.20.1



Slika 9.11. *Mrežna topologija za primjer VLAN konfiguracije s adresama – zadatak 2*

Potrebni su nam:

- 2 preklopnika (Cisco 2960)
- 4 PC-a
- Kabeli
- Usmjerivač (Cisco 1941)

KREIRANJE VLAN-ova na preklopnicima

Kreiranje vlan-ova na S1

```
Switch>enable
Switch#configure terminal
Switch(config)#no ip domain-lookup
Switch(config)#hostname S1

S1(config)#vlan 10
S1(config-vlan)#name student
S1(config-vlan)#exit

S1(config)#vlan 20
S1(config-vlan)#name faculty
S1(config-vlan)#exit
S1(config)#
```

Koristeći naredbu *show vlan* na S1 što možete primjetiti?

```
S1#show vlan

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                 Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                 Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                 Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                 Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                 Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                 Gig0/1, Gig0/2
10   student                           active
20   faculty                           active
1002 fddi-default                      act/unsup
1003 token-ring-default                act/unsup
1004 fddinet-default                   act/unsup
1005 trnet-default                     act/unsup

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        0      0
10   enet  100010     1500  -      -      -        -    -        0      0
```

Slika 9.12. *Prikaz ispisa nakon naredbe „Show vlan“ na S1*

VLAN-ovi su kreirani, ali još nemaju niti jedno pridruženo sučelje!

Kreiranje VLAN-ova na S2

```
Switch>enable
Switch#configure terminal
Switch(config)#no ip domain-lookup
Switch(config)#hostname S2

S2(config)#vlan 10
S2(config-vlan)#name student
S2(config-vlan)#exit

S2(config)#vlan 20
S2(config-vlan)#name faculty
S2(config-vlan)#exit
S2(config)#
```

```
S2#sh vlan brief

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                 Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                 Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                 Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                 Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                 Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                 Gig0/1, Gig0/2
10   student                           active
20   faculty                           active
1002 fddi-default                      active
1003 token-ring-default                active
1004 fddinet-default                   active
1005 trnet-default                     active
S2#
```

Slika 9.13. *Prikaz ispisa nakon naredbe „Sh vlan brief“ na S2*

Oglašavanje VLAN-a na željena sučelja *(port* ili *interface*)

U našem zadatku kod **S1** žeimo da sučelja fa0/2, fa0/3, fa0/4 pripadaju vlan-u 20, a sučelja 6-24 ćemo postavit u vlan 10.

Na **S2** ćemo sučelja fa0/2, fa0/3, fa0/4 i fa0/5 postaviti da pripadaju vlan-u 20, a sučelja 6-24 ćemo postavit u vlan 10.

Oglašavanje vlana na točno određena sučelja kod S2

Sučelja 2-5 ćemo dodijeliti vlan 20 na sljedeći način:

```
S2(config)#interface range fastEthernet 0/2-5
S2(config-if-range)#switchport mode access
S2(config-if-range)#switchport access vlan 20
S2(config-if-range)#end
S2#
```

Sučelja 6-24 ćemo dodijeliti vlan 10 na sljedeći način:

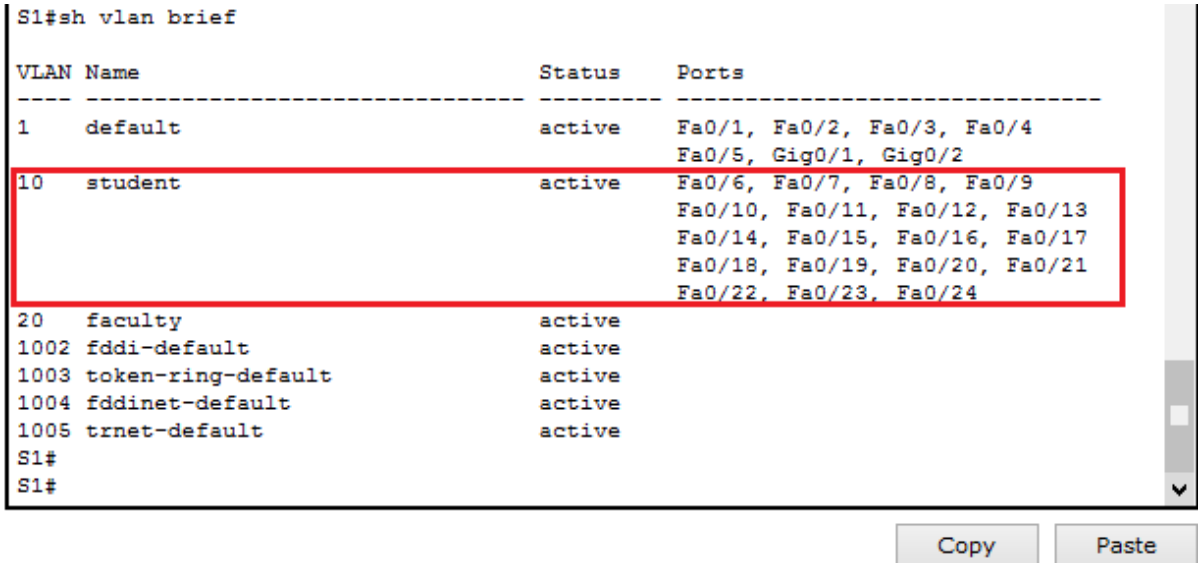
```
S2(config)#interface range fastEthernet 0/6-24
S2(config-if-range)#switchport mode access
S2(config-if-range)#switchport access vlan 10
S2(config-if-range)#end
S2#show vlan brief       //što primjećujete?
```

```
S2#show vlan brief

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/1, Gig0/1, Gig0/2
10   student                           active    Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                 Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                 Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                 Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                 Fa0/22, Fa0/23, Fa0/24
20   faculty                           active    Fa0/2, Fa0/3, Fa0/4, Fa0/5
1002 fddi-default                      active
1003 token-ring-default                active
1004 fddinet-default                   active
1005 trnet-default                     active
S2#
```

Slika 9.14. *Prikaz ispisa nakon naredbe „Show vlan brief“ nakon dodjele sučelja vlan 10*

Oglašavanje vlana na točno određena sučelja kod S1

```
S1(config)#interface range fa0/6-24
S1(config-if-range)#switchport mode access
S1(config-if-range)#switchport access vlan 10
S1(config-if-range)#end
S1#sh vlan brief
```

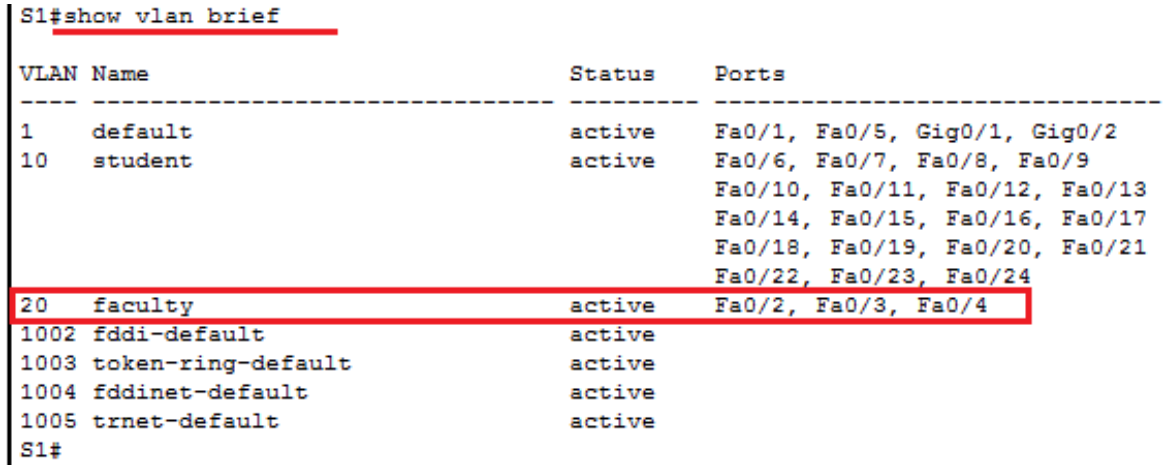


```
S1#sh vlan brief

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                 Fa0/5, Gig0/1, Gig0/2
10   student                           active    Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                 Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                 Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                 Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                 Fa0/22, Fa0/23, Fa0/24
20   faculty                           active
1002 fddi-default                      active
1003 token-ring-default                active
1004 fddinet-default                   active
1005 trnet-default                     active
S1#
S1#
```

Copy Paste

Slika 9.15. *Prikaz ispisa nakon naredbe „Show vlan brief“ nakon dodjele sučelja vlan 10*

```
S1(config)#interface range fastEthernet 0/2-4
S1(config-if-range)#switchport mode access
S1(config-if-range)#switchport access vlan 20
S1(config-if-range)#end
```



```
S1#show vlan brief

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/1, Fa0/5, Gig0/1, Gig0/2
10   student                           active    Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                 Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                 Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                 Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                 Fa0/22, Fa0/23, Fa0/24
20   faculty                           active    Fa0/2, Fa0/3, Fa0/4
1002 fddi-default                      active
1003 token-ring-default                active
1004 fddinet-default                   active
1005 trnet-default                     active
S1#
```

Slika 9.16. *Prikaz ispisa nakon naredbe „Show vlan brief“ nakon dodjele sučelja vlan 20*

Konfiguracija preklopnika i usmjerivača (postavljanje u *trunk mode* i enkapsulacija)

Konfiguracija preklopnika S1

```
S1(config)#int fa0/1
S1(config-if)#switchport mode trunk          -taj link stavlja u trunk mode
S1(config-if)#switchport trunk allowed vlan all
S1(config)#interface fastEthernet 0/5
S1(config-if)#switchport mode trunk
S1(config-if)#switchport trunk allowed vlan all
S1(config-if)#exit
S1(config)#
```

Konfiguracija preklopnika S2

```
S2(config)#interface fa0/1
S2(config-if)#switchport mode trunk
S2(config-if)#switchport trunk allowed vlan all
```
- možemo biti i preciziniji pa napisati

switchport trunk allowed vlan 10,20,1002-1005

Zašto smo prvo razdvojili računala VLAN-ovima da ne mogu komunicirati, a sada im spajanjem preko usmjerivača opet omogućavamo komunikaciju?

Razlog 1 - Kao što vidimo, računala se nalaze na **različitim mrežama** (192.168.10.0/24 i 192.168.20.0 /24), pa ih bez kreiranja VLAN-a ne bi mogli sve pospojiti na jedan preklopnik. Različite mreže međusobno se povezuju usmjerivačima, a ne preklopnicima. Na preklopnik se u normalnom slučaju spajaju samo računala koja pripadaju istoj mreži.

Razlog 2– Istu funkcionalnost mogli smo dobiti bez kreiranja VLAN-a, na način da za svaku mrežu imamo zasebni preklopnik, a zatim svaki preklopnik povežemo na zasebno sučelje usmjerivača. U ovome primjeru kreirali smo samo 2 VLAN-a. To je zbog vremenskog trajanja vježbe. Međutim, u realnom životu imamo potrebe za npr. 10 VLAN-ova. Ovim načinom koristimo znatno manji broj uređaja što je **znatno jeftinije!!!**

Razlog 3 – Komunikaciju između VLAN-ova smo ponovno omogućili, ali ovaj put preko usmjerivača na kojem možemo primjeniti razne sigurnosne opcije

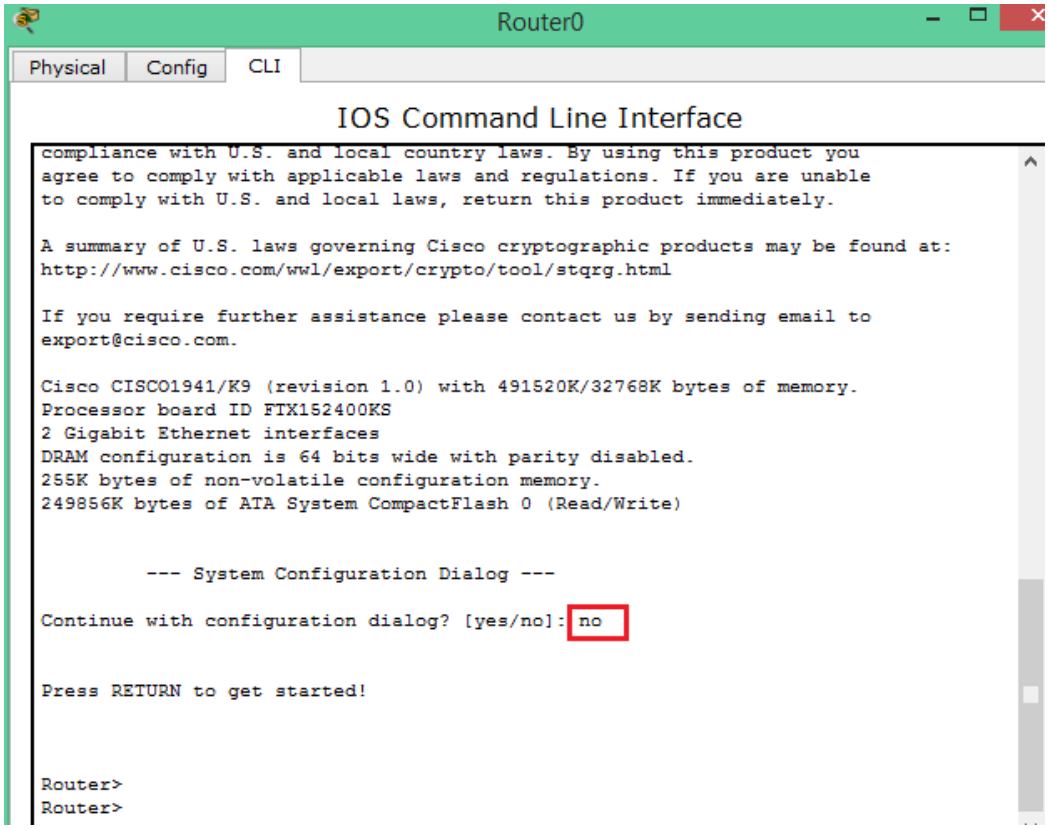
Da rekapituliramo:

Prvo smo razdvojili jednu veliku mrežu u više VLAN-ova da bismo smanjili „*broadcast domenu*“ tj. količinu broadcast poruka koja prolazi mrežom i kako bi podigli sigurnost.

Međutim, mi bismo ipak htjeli da sva računala mogu komunicirati, ali na sigurniji način. Kako bismo to ostvarili, sve smo ih međusobno povezali preko usmjerivača. Kako usmjerivači na svakom svom sučelju moraju imati različitu mrežu, nužno je da svaki VLAN bude u različitim mrežama.

Obzirom da su sučelja usmjerivača skupa, izmišljena je „enkapsulacija dot1q“ kojom se kroz jedno fizičko sučelje može kreirati više virtualnih sučelja, pa cijelu cijelu korporacijsku mrežu (višestruke VLAN-ove) možemo povezati preko samo jednog sučelja usmjerivača. (za njihovu međusobnu komunikaciju, ali i za izlaz prema javnoj mreži)

Konfiguracija usmjerivača



Slika 9.17. *Početak konfiguracije dot1q*

Konfiguracija dot1Q metode spajanja VLAN-ova

```
R1(config)#interface gigabitEthernet 0/1.10
R1(config-subif)#encapsulation dot1Q 10
R1(config-subif)#ip address 192.168.10.1 255.255.255.0
R1(config-subif)#no shutdown
R1(config-subif)#exit
R1(config)#

R1(config-subif)#interface gigabitEthernet 0/1.20
R1(config-subif)#encapsulation dot1Q 20
R1(config-subif)#ip address 192.168.20.1 255.255.255.0
R1(config-subif)#no shutdown
R1(config-subif)#exit

R1(config)#interface gigabitEthernet 0/1
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#
```

Isto fizičko sučelje dobiva dvostruku IP adresu!!! Jer kroz njega idu paketi za dvije mreže!!!

Moguće napraviti jer ga sa dot1Q enkapsulacijom logički podijelimo!

Objašnjenje:

`interface gigabitEthernet 0/1.10` - .10 predstavlja vlan koji radimo

`encapsulation dot1Q 10` - govori iz kojeg će vlan-a uzimati pakete

Samostalni zadatak:

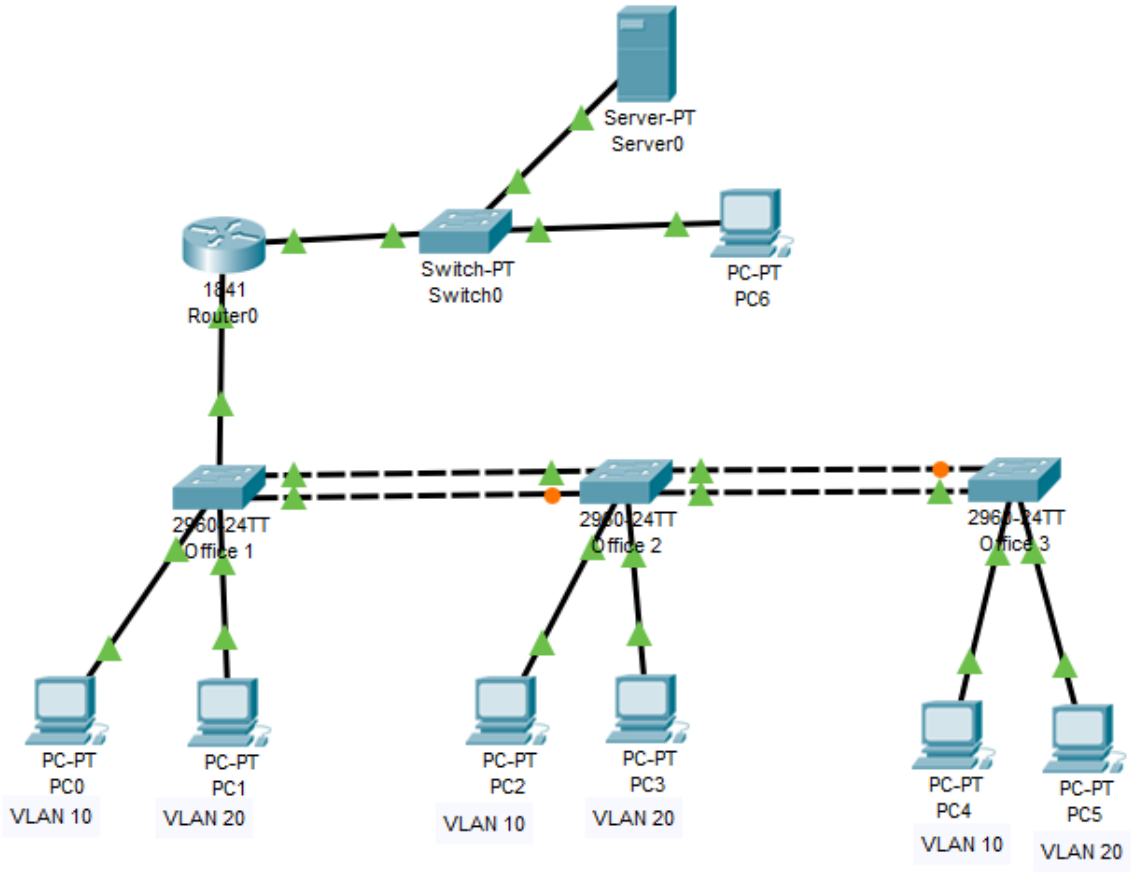
Pokušajte sada samostalno riješiti slijedeći zadatak. Riješenja se nalaze na slijedećoj stranici.

Na Moodle-u možete pronaći početnu topologiju zadatka u CPT, kao i konačnu potpuno konfiguriranu mrežu. (Vježba 10a)

Zadatak:
Imamo urede na tri kata, na svakome katu imamo računala koja su podijeljena u 2 VLAN-a.

- VLAN 10 - adrese 10.0.0.0/24
- VLAN 20 - adrese 20.0.0.0/24

Konfiguriraj VLAN-ove, međusobno ih poveži preko usmjerivača „dot1Q“ metodom, te omogući spoj na poslužitelj i računalo u mreži 15.0.0.0/24



Slika 9.18. *Zadana mrežna topologija za Spanning-Tree zadatak*

Dvostruka trunk linija između preklopnika postoji radi sigurnosti. Radi samo jedna, a druga je u "*hot standby*" modu (stalno svijetli narančasto) i aktivira se automatski u slučaju ispada radne linije. Ovu finkcionalnost kontrolira „spanning-tree“ protokol – u ovom slućaju nije potrebno ništa konfigurirati

TEST: Nakon konfiguracije cijele mreže i provjere funkcionalnosti izbriši radnu trunk liniju i pričekaj nekoliko sekundi, te vidi kako će druga linija preuzeti rad.

Rješenje:

Konfiguracija usmjerivača

```
Router(config)#interface fastEthernet 0/0
Router(config-if)#no shutdown
Router(config)#interface fastEthernet 0/0.10
Router(config-subif)#encapsulation dot1q 10
Router(config-subif)#ip address 10.0.0.1 255.255.255.0
Router(config-subif)#exit
Router(config)#interface fastEthernet 0/0.20
Router(config-subif)#encapsulation dot1q 20
Router(config-subif)#ip address 20.0.0.1 255.255.255.0
Router(config-subif)#exit
```

Sami kreirajte sučelje prema poslužitelju

Konfiguracija preklopnika S1 (ostale napravite sami)

```
s1(config)#vlan 10
s1(config-vlan)#name proizvodnja
s1(config-vlan)#exit
s1(config)#vlan 20
s1(config-vlan)#name uprava
s1(config-vlan)#exit

s1(config)#interface fastEthernet 0/1
s1(config-if)#switchport mode access
s1(config-if)#switchport access vlan 10
s1(config-if)#exit
s1(config)#interface fastEthernet 0/2
s1(config-if)#switchport mode access
s1(config-if)#switchport access vlan 20
s1(config-if)#exit
s1(config)#

s1(config)#interface gigabitEthernet 0/2
s1(config-if)#switchport mode trunk
s1(config-if)#switchport trunk allowed vlan all
s1(config-if)#exit

s1(config)#interface fastEthernet 0/24
s1(config-if)#switchport mode trunk
s1(config-if)#switchport trunk allowed vlan all
```

10. WAN (*Wide Area Network*) i PPP (*Point to Point*) protokol

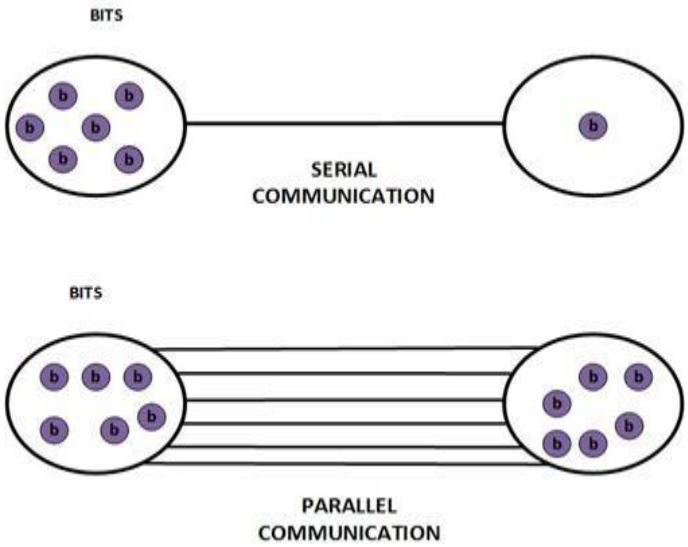
U dosadašnjem dijelu bavili smo se problematikom lokalnih mreža. Međutim, razne tvrtke imaju svoje urede na udaljenim lokacijama te je potrebno izvršiti međusobno povezivanje. Tu na scenu stupa WAN.

Osnovne razlike između WAN i LANs mreža su:

- Geografske udaljenosti. WAN mreže povezuju znatno veće udaljenosti
- Upravljanje WAN mrežama uglavnom se vrši od strane neke telekomunikacijske tvrtke, a tvrtke korisnici plaćaju telekomunikacijskim tvrtkama usluge korištenja WAN mreže.
- U LAN mrežama možemo koristiti paralelne veze između uređaja, dok kod WAN veza uvijek koristimo serijske veze obzirom da se protežu preko velikih udaljenosti.

Razlika između serijske i paralelne veze

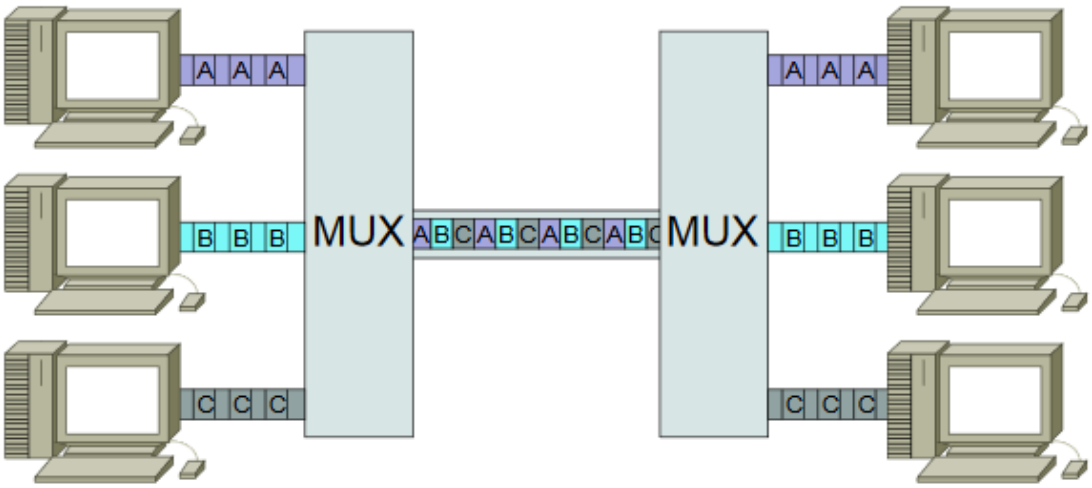
Slika ispod prikazuje razliku između serijske i paralelne veze.



Slika 10.1. *Serijska i paralelna komunikacija*

Kod serijske veze informaciju prenosimo bit po bit dok kod paralelne komunikacije istovremeno šaljemo više bita što dovodi do toga da je paralelne komunikacija učinkovitija. Međutim, paralelna komunikacija zahtjeva da svi biti stignu istovremeno na odredište, što kod velikih udaljenosti predstavlja problem. Paralelna komunikacija zahtjeva veći broj žica, što znatno poskupljuje izvedbu same veze (kabel ima više žica pa je skuplji). Unutar kabela sa više žica na velikim udaljenostima dolazi do preslušavanja usljed savijanja kabela. Iz svih tih razloga, za premošćivanje velikih udaljenosti znatno su efikasnije serijske veze jer u konačnici mogu raditi na znatno višim frekvencijama. Serijski komunikacijski kanal moguće je dijeliti između više uređaja prepletanjem signala podjelom vremena tj. vremenskim multipleksiranjem (*Time-Division Multiplexing - TDM*). TDM prepletanje omogućuje prijenos više signala, ili podataka iz više izvora, preko zajedničkog komunikacijskog kanala te

rekonstrukciju izvornih podataka na odredištu. U primjeru na slici ispod prikazano je prepletanje triju signala.



Slika 10.2. *Vremensko prepletanje ili „multipleksiranje“ [1]*

Postupak prepletanja provodi se odabirom dijela ulaznih signala. To je najčešće jedan bit ili jedan oktet (*byte*) svakog signala pa se TDM postupci dijele na:

- one koji prepleću bitove (*bit-interleaving*) i
- one koji prepleću oktete (*byte-interleaving*).

Vremenski intervali za prijenos dijelova signala iz svih izvora prisutni su neovisno o tome jesu li spremni novi podaci za slanje. U primjeru na slici šalju se isprepleteni podaci iz tri izvora, podaci iz prvog izvora označeni su s A, iz drugog slovom B, a iz trećeg izvora slovom C. Ako su na ulazu prisutni podaci sa sva tri izvora podaci se šalju redom: A, B, C. Ako u jednom trenutku nema novih podataka iz drugog izvora redoslijed slanja podataka je: A, „_“, C, gdje je oznakom „_“ naznačen vremenski period u kojem se ne šalju podaci. TDM se provodi na fizičkom sloju, neovisno o tipu podataka koji se šalju ili protokolu koji se izvodi na podatkovnom sloju. Primjer ovakvog ispreplitanja je ISDN (*Integrated Service Digital Network*) sustav s dva kanala B1 i B2 brzine prijenosa 64 kbps i jedan D kanal brzine prijenosa 16 kbps. Prijenos se provodi u devet intervala koji se ponavljaju, pa je slijed slanja: ... B1, B2, B1, B2, B1, B2, B1, B2, D,

WAN Enkapsulacijski protokoli

U WAN okruženju moramo odrediti enkapsulacijski protokol kako bi bili sigurni da će okviri koji šaljemo sa jedne strane biti ispravno ponovno sastavljeni na drugoj strani. Naime, serijska veza je ništa više nego „mašina“ koja provodi TDM multipleksiranje, bez neke posebne „pameti“. Obzirom da okviri iz različitih izvora koji stižu na odašiljačku stranu serijske veze mogu biti različitih veličina, prijamna strana nezna kada je okvir u potpunosti pristigao, te ga mora proslijediti na obradu višim slojevima. Zato je neophodno da se kroz serijsku vezu, osim samih korisnih podataka, propušta i neki protokol koji će osigurati potrebnu „pamet“. Postoji više enkapsulacijskih protokola (Frame Relay, HDLC,..), ali u ovom kolegiju obrađivati ćemo samo PPP protokol.

U CPT-u već smo koristili serijske veze između uređaja i sve je funkcioniralo bez bilo kakve dodatne konfiguracije iz razloga što je Cisco usmjerivačima već po *defaultu* postavljeno korištenje HDLC protokola na svim serijskim sučeljima.

PPP protokol (*Point-to-Point Protocol*)

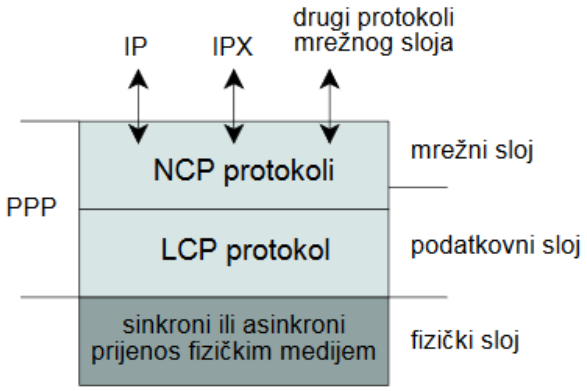
Point-to-Point Protocol (PPP) je „*open standard protocol*“ koji se najčešće koristi za uspostavu **direktne serijske** veze između dva čvora. Najčešči slučaj je priključivanje računala pomoću serijskog kabla, telefonske linije, optičkih vlakana, mobilnih telefona ili UTP mrežnih kablova. Glavna uloga PPP-a je prijenos paketa trećeg sloja kroz drugi sloj *point-to-point* veze obzirom da se IP paketi trećeg sloja ne mogu odašiljati kroz neki kanal (npr. modem) drugog sloja bez nekog popratnog protokola drugog sloja koji će uspostaviti i održavati vezu, a zatim se okviri PPP protokola dodatno enkapsuliraju u okvire prvog sloja (praćenje i otklanjanje pogrešaka, dogovor o brzini prijenosa, itd.). Većina Internet kompanija koriste PPP za *dial-up* pristup internetu. Također je moguć PPP preko Etherneta (PPPoE), povezujući neki DSL modem sa računalom preko mrežne kartice što je brže nego povezivanje sa USB ili nekom drugom sabirnicom. PPP protokol uglavnom je izbacio iz uporabe starije protokole koji su se prije koristili za tu namjenu poput LAPB (iz porodice X.25 protokola).

PPP može biti konfiguriran na:

- Asinhronoj serijskoj vezi (kao što je *Plain old telephone service - POTS)*
- Sinhronoj serijskoj vezi (poput ISDN-a ili *point-to-point* zakupljene linije).

PPP se sastoji od dva pod-protokola:

- ***Link Control Protocol*** *(LCP*): uspostava veze i pregovaranje o kontrolnim postavkama na drugom sloju OSI modela. Nakon uspješne uspostave veze započinje uporaba NCP.
- ***Network control Protocol*** *(NCP*): obavlja pregovaranje o opcijskim konfiguracijskim parametrima, te priprema sve neophodno za normalno funkcioniranje trećeg sloja. Drugim riječima rečeno, osigurava da IP i ostali protokoli mogu normalno funkcionirati kroz PPP vezu

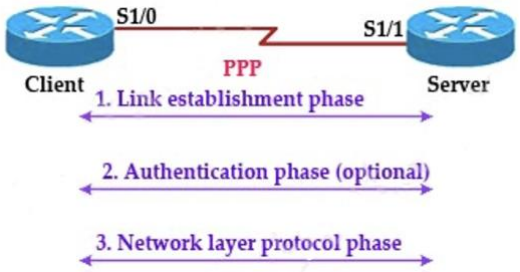


Slika 10.3. *Slojevita struktura PPP protokola [1]*

Uspostava PPP sjednice

Veza mora proći kroz 3 faze uspostave PPP sjednice:

1. **Faza uspostave veze (***Link establishment phase***)**: U ovoj fazi, svaki PPP uređaj šalje LCP pakete kako bi se konfigurirala i testirala veza na drugom sloju.
2. **Autentifikacijska faza (***Authentication phase)* – nije obavezna: Ukoliko želimo postaviti autentifikaciju pri uspostavi veze tada ćemo koristiti PAP ili CHAP autentifikacijske protokole koji se koriste za uporabu u PPP vezama.
3. **Faza uspostave mrežnog protokolnog sloja (***Network layer protocol phase***)**: PPP šalje NCP pakete kako bi odabrao i konfigurirao enkapsulaciju i slanje mrežnih protokola trećeg sloja kroz PPP vezu



Slika 10.4. *Faze uspostave veze kod PPP protokola*

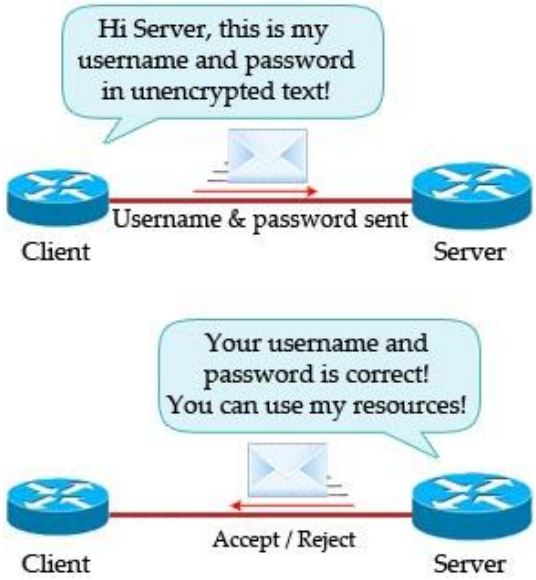
Pazi: Standardna način serijske enkapsulacije na Cisco usmjerivačima je HDLC, a ne PPP. Ako želite koristiti PPP – morate ga konfigurirati. Za razliku od HDLC-a koji je Ciscov protokol, PPP je „*open standard*“ protokol pa ga treba koristiti pri spajanju Cisco usmjerivača sa usmjerivačima ostalih proizvođača.

PPP autentifikacijske metode

Postoje dvije autentifikacijske metode koje se koriste u PPP protokolu:

- *Password Authentication Protocol* (PAP)
- *Challenge Handshake Authentication Protocol (*CHAP)

Password Authentication Protocol **(PAP)** je vrlo jednostavan autentifikacijski protokol. Klijentski uređaj koji želi uspostaviti vezu sa serverom šalje ime (*username*) i lozinku (*password*) u obliku čistog teksta. („Klijent“ je uređaj koji prvi započinje sa komunikacijom, a „server“ je onaj koji odgovara – ovdje je riječ o sučeljima na usmjerivačima!!). Server provjerava ispravnost imena i lozinke, te ovisno o tome prihvaća ili odbacuje komunikaciju. Ovo se naziva „dvostruko rukovanje“ (*two-way handshake*). U PAP-ovom dvostrukom rukovanju, ime i lozinka šalju se odmah u prvoj poruci.



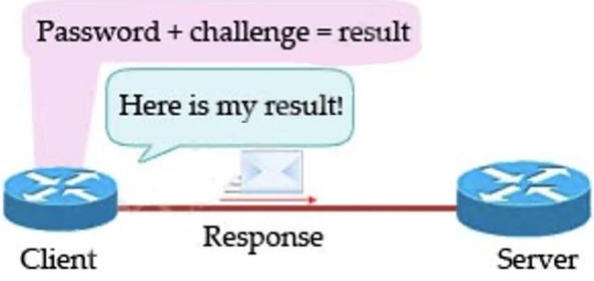
Slika 10.5. *PAP „dvostruko rukovanje“*

Za sustave koji trebaju veću razinu sigurnosti PAP nije dovoljan jer ukoliko još netko ima pristup vezi, može vrlo jednostavno doći do imena i lozinke, a nakon toga i pristupiti resursima. U tom slučaju koristi se CHAP!

Challenge Handshake Authentication Protocol (CHAP) je PPP autentifikacijski protokol koji je znatno sigurniji od PAP-a. CHAP koristi „trostruko rukovanje“:

Slika 10.6. *CHAP „challenge“*

CHAP protokol započinje sa nasumičnom tekstualnom porukom zvanom„*challenge*“koja se šalje od servera prema klijentu kako bi se on autentificirao.



Slika 10.7. *CHAP „response“*

Nakon primitka„*challenge*“ poruke, klijent koristi svoju lozinku za enkripciju teksta iz „*challenge*“ poruke algoritmom MD5, te je tako enkriptiranu vraća serveru. Čak ako netko i presretne tu poruku, neće iz nje moći saznati lozinku.

Slika 10.8. *CHAP „Accept/reject“*

Server vrši isti proces. Ukoliko je klijentov odgovor isti kao i serverov rezultat, lozinka je ispravna. Znači, glavna razlika između PAP i CHAP je u tome što PAP šalje ime i lozinku kroz vezu, a kod CHAP-a, lozinka se nikada ne šalje kroz vezu. Druga razlika između ova dva autentifikacijske procesa je u tome što PAP vrši autentifikaciju samo pri inicijalnoj uspostavi veze, dok CHAP još to radi i periodično. Tekst koji se koristi za „*challenge*“poruke je prilikom svake autentifikacije potpuno nasumičan kako bi se spriječili tzv. „*playback*“ napade kod kojih hakeri pokušavaju kopirati „odgovor“ koji klijent šalje nazad prema serveru i ponovno ga koristiti.

PAP i CHAP konfiguracija

Konfiguracija PAP i CHAP protokola vrlo je jednostavna. Prvo moramo omogućiti PPP enkapsulaciju, a zatim specificirati koji autentifikacijski protokol će se koristiti (PAP ili CHAP). Naredbe za to su “ppp authentication pap” ili “ppp authentication chap”.

PAP Konfiguracija

U mnogim knjigama naći ćete kako dva usmjerivača autentificiraju jedan drugoga i njihove konfiguracije su potpuno identične. Međutim, kako bi bolje razumjeli proces i razliku između klijenta i servera, u ovom primjeru napraviti ćemo da samo server autentificira klijenta, a ne i obratno.

Slika 10.9. *PAP konfiguracija*

```
Client(config)#int s1/0
Client(config-if)#encapsulation ppp
Client(config-if)#ppp pap sent-username CLIENT1 password TUT
Client(config-if)#no shutdown
```

```
Server(config)#username CLIENT1 password TUT
Server(config)#int s1/1
Server(config-if)#encapsulation ppp
Server(config-if)#ppp authentication pap
Server(config-if)#no shutdown
```

Prvo moramo omogućiti PPP enkapsulaciju na sučeljima oba usmjerivača naredbom "encapsulation ppp".

Serverski usmjerivač je onaj koji će vršiti autentifikaciju nakon što primi ime i lozinku od klijenta, pa ćemo njega konfigurirati naredbom "ppp authentication pap".

U serverskom usmjerivaču također moramo definirati zapis imena i lozinke koji će trebati biti pogođeni od strane klijetskog usmjerivača. Taj zapis postavljamo naredbom "username CLIENT1 password TUT" command.

Možete vidjeti da smo u klijentovoj konfiguraciji kao „ime" koristili drugačije naziv od naziva uređaja!!! („CLIENT1" umjesto „Client") .

Ukoliko je vaša konfiguracija ispravna pojaviti će se zelena oznaka na sučeljima koja označava vezu u radu.

Pazi: Ne koristite naredbu "ppp authentication pap" na klijentskom usmjerivaču jer ne želimo da klijent autentificira serverski usmjerivač. Ukoliko koristite tu naredbu PPP veza neće raditi jer serverski usmjerivač nije konfiguriran da šalje ime i lozinku prema klijentu!

CHAP konfiguracija

CHAP konfiguracija je vrlo slična kao i PAP.

```
Client(config)#interface Serial 1/0
Client(config-if)#encapsulation ppp
Client(config-if)#ppp chap hostname CLIENT1
Client(config-if)#ppp chap password TUT
Client(config-if)#no shutdown

Server(config)#username CLIENT1 password TUT
Server(config)#interface Serial 1/1
Server(config-if)#encapsulation ppp
Server(config-if)#ppp authentication chap
Server(config-if)#no shutdown
```

Pazi: Ne koristite naredbu "ppp authentication chap" na klijentskom usmjerivaču jer ne želimo da klijent autentificira serverski usmjerivač. Ukoliko koristite tu naredbu PPP veza neće raditi jer serverski usmjerivač nije konfiguriran da šalje ime i lozinku prema klijentu!

Provjera enkapsulacije na serijskom sučelju

Možemo koristiti "show interface <interface>" naredbu kako bi vidjeli konfiguriranu enkapsulaciju na serijskom sučelju, te status LCP i NCP ukoliko je PPP enkapsulacija konfigurirana.

```
Client#show interface s1/0
Serial1/0 is up, line protocol is up
  Hardware is M4T
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
Encapsulation PPP, LCP Open
  Open: CDPCP, crc 16, loopback not set
```

Možemo vidjeti da je na sučelju Serial1/0 konfigurirana PPP enkapsulacija. LCP stanje je "*open*" što znači da pregovaranje i uspostava veze bila uspješna. "*Open: CDPCP*" linija nam govori da NCP protokol osluškuje Cisco Discovery Protocol (CDP) protokol.

Pazi: osim PAP i CHAP u PPP protokol nedavno je uvedena i treća autentifikacijska metoda tzv.***Extensible Authentication Protocol (EAP)*** koju nećemo obrađivati u ovom kolegiju.

11.NAT (*Network Address Translation*)

Kada komuniciramo sa nekim uređajem u javnoj mreži, naš uređaj mora koristiti adresu izvora informacije koja pak mora biti javna IP adresa. NAT uređaj omogućava privatnim IPv4 adresama da se spoje na Internet gdje se koriste isključivo javne IP adrese. NAT uređaj nam omogućuje zamjenu IP adrese izvorišta informacije iz IP paketa, u neku drugu adresu. Uobičajeno, NAT spaja dvije mreže i translatira (prijevodi) privatne adresu u javne prije nego ti paketi napuste našu privatnu mrežu.

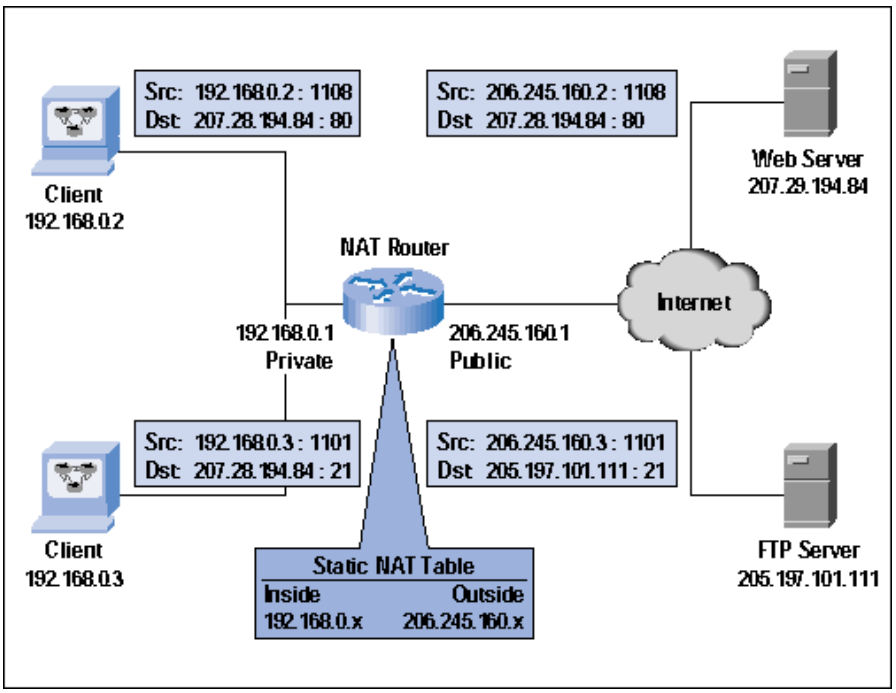
Situacije gdje koristimo NAT

- ISP nam nije dodijelio dovoljan broj javnih IP adresa
- Naša kompanija će se spojiti sa nekom drugom koja koristi isti adresni prostor
- Kada želimo sakriti naše unutarnje IP adrese od vanjskog svijeta
- Kada želimo dodjeliti istu IP adresu višestrukim uređajima

Postoje 3 vrste NAT-a:

- Statički
- Dinamički
- PAT

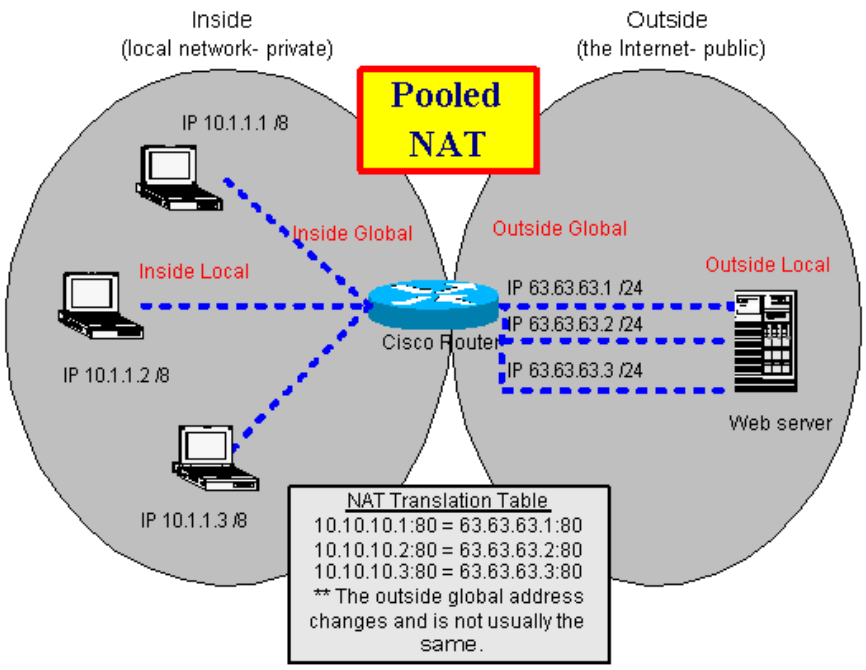
Statički NAT



Slika 11.1. *Statički NAT*

Kod statičkog NAT-a manualno konfiguriramo uređaj koji provodi NAT postupak, prevođenja jedne IP adrese u drugu. Ukoliko imamo 100 uređaja, trebati će nam 100 statičkih unosa u tablicu adresne translacije.

Dinamički NAT (*Pooled NAT*)



***Slika 11.2.** Dinamički NAT*

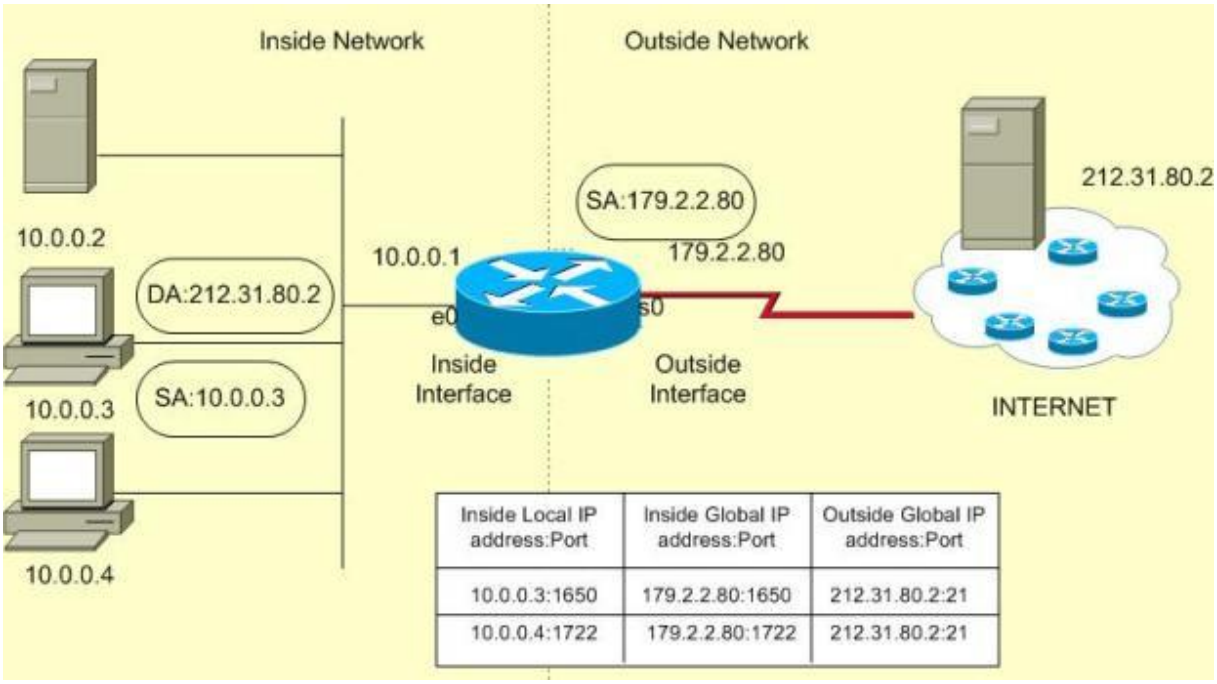
Dinamički NAT uglavnom se upotrebljava kada „unutarnji“ korisnici žele pristupati vanjskim uređajima i sadržajima. Javna adresa koja se dodjeljuje unutarnjem korisniku je nebitna jer vanjski uređaj (npr. Web server) samo vraća zatražene podatke na adresu sa koje je stigao upit. Kada unutarnji korisnik šalje podatke kroz NAT uređaj (npr. usmjerivač), tada taj NAT uređaj prvo ispita izvorišnu IP adresu iz paketa te je uspoređuje bazom lokalnih privatnih IP adresa (onih kojima je dozvoljeno da izlaze van kroz NAT uređaj).

Ako pronađe preklapanje, tada određuje iz kojeg seta javnih adresa će mu dodjeliti neku javnu adresu. Naime, NAT uređaju možemo dodjeliti nekoliko setova javnih adresa (npr. jedan set za korisnike iz prodaje, jedan set za korisnike iz uprave, jedan set za korisnike iz proizvodnog pogona, itd.) Nakon određivanja seta javnih adresa, NAT uređaj **dinamički** uzima jednu adresu iz tog seta koja **trenutno** nije dodjeljena ni jednom drugom uređaju, te je dodjeljuje. Usmjerivač upisuje tu vrijednost u svoju adresno translacijsku tablicu, te zamjenjuje izvorišnu (privatnu) IP adresu dodjeljenom javnom, te prosljeđuje paket dalje u vanjski svijet.

Ukoliko usmjerivač ne pronađe preklapanje izvorišne IP adrese sa njegovim setovima lokalnih adresa, tada neće doći do translacije adrese, već će paket biti proslijeđen u vanjski svijet u svom originalnom obliku.

Kada povratni promet iz vanjskog svijeta dolazi natrag u mrežu, NAT uređaj provjerava odredišnu IP adresu i uspoređuje je sa svojom tablicom adresnih translacija. Nakon što pronađe podudaranje, translatira globalnu adresu u lokalnu, vrši zamjenu odredišne adrese u IP paketu, te je prosljeđuje u lokalnu mrežu.

PAT (*Port Address Translation*)



***Slika 11.3.** PAT*

Kod PAT-a, svi uređaji koji prolaze kroz uređaj za translaciju adsresa dobiju istu globalnu IP adresu, ali za razlikovanje različitih **veza** (ne uređaja, jer jedno računalo može istovremeno imati više različitih sjednica) koriste se različiti brojevi TCP ili UDP porta. Ukoliko dva različita uređaja imaju isti broj porta, tada NAT uređaj promjeni jednog od njih kako bi osigurao jedinstvenost. Glavna razlika između NAT i PAT je u tome da se kod NAT-a mjenja isključivo IP adresa, (ne i broj porta).

Nedostaci translacije adresa

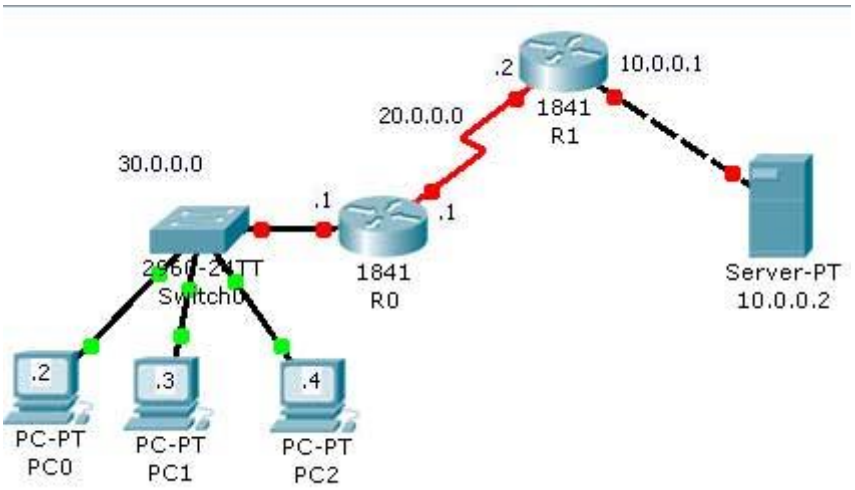
Postoje tri glavna nedostatka:

- Svakoj vezi unosi određeno kašnjenje
- Otkrivanje mogućih problema je kompliciranije
- Neke aplikacije ne mogu raditi uz adresne translacije

Zadatak - Konfiguracija statičkog NAT-a

Konfiguracija statičkog NAT-a je vrlo jednostavna. U slijedećem primjeru imamo web poslužitelj (*server*) spojen sa usmjerivačem Router 1. Naš web poslužitelj koristi IP adresu 10.0.0.2. ali iz nekog razloga (gore diskutirani) firma želi koristiti adresu 50.0.0.1 za ovaj poslužitelj. Naša je zadaća konfigurirati NAT na usmjerivaču R1 tako da prevodi (translatira) 10.0.0.2 [unutarnju lokalnu adresu web poslužitelja] u 50.0.0.1 [unutarnju globalnu IP adresu].

U CPT nacrtajte slijedeću topologiju



Slika 11.4. *Mrežna topologija za NAT - zadatak 1*

Postavite IP adrese na sučelja računala i poslužitelja kako je označeno na slici.

Konfigurirajte R1 slijedeći upute ispod

```
Router>enable
Router#configure terminal
Router(config)#hostname R1
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 20.0.0.2 255.0.0.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface fastEthernet 0/0
R1(config-if)#ip nat inside - definicija „unutarnje“ mreže na ovom sučelju
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip nat outside - definicija „vanjske“ mreže na ovom sučelju
R1(config-if)#exit
R1(config)#ip route 30.0.0.0 255.0.0.0 20.0.0.1 - postavljanje statičkog usmjeravanja
R1(config)#ip nat inside source static 10.0.0.2 50.0.0.1 - postavljanje statičke NAT
translacije adrese)
R1(config)#
```

Sada konfigurirajte R0 slijedeći upute ispod

```
Router>enable
Router#configure terminal
Router(config)#hostname R0
R0(config)#interface fastethernet 0/0
R0(config-if)#ip address 30.0.0.1 255.0.0.0
R0(config-if)#no shutdown
R0(config-if)#exit
R0(config)#interface serial 0/0/0
R0(config-if)#ip address 20.0.0.1 255.0.0.0
R0(config-if)#clock rate 64000
R0(config-if)#bandwidth 64
R0(config-if)#no shutdown
R0(config-if)#exit
R0(config)#ip route 50.0.0.0 255.0.0.0 20.0.0.2 - postavljanje statičkog usmjeravanja
R0(config)#
```

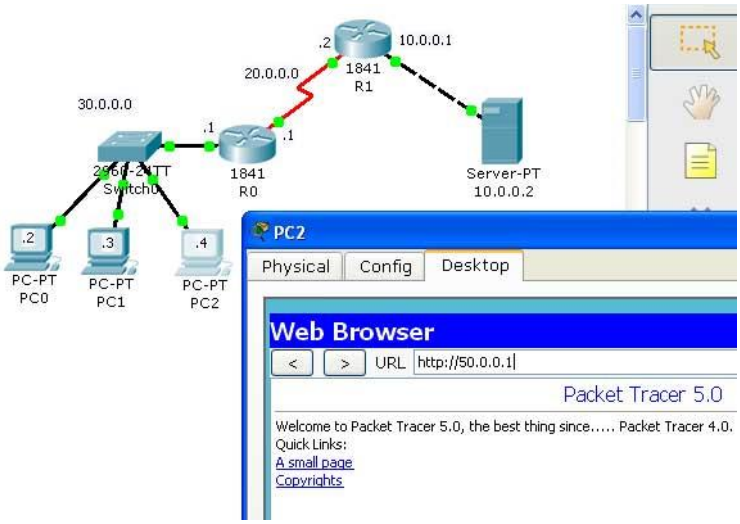
Kao što vidite u konfiguraciji, nema direktne rute prema 10.0.0.2. pa računala iz mreže 30.0.0.0 nikada neće znati za postojanje te adrese. Računala će pristupati adresi 50.0.0.1 kada žele pristupiti web poslužitelju. Za provjeru konfiguracije, sa bilo kojeg računala pošaljite „ping“ naredbu na adresu 50.0.0.1 i dobiti ćete odgovor.

```
Packet Tracer PC Command Line 1.0
PC>ping 50.0.0.1
Pinging 50.0.0.1 with 32 bytes of data:
Reply from 50.0.0.1: bytes=32 time=141ms TTL=126
Reply from 50.0.0.1: bytes=32 time=80ms TTL=126
Reply from 50.0.0.1: bytes=32 time=109ms TTL=126
Reply from 50.0.0.1: bytes=32 time=125ms TTL=126

Ping statistics for 50.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 80ms, Maximum = 141ms, Average = 113ms
```

Sada pokušajte poslati ping naredbu na 10.0.0.2 i dobiti ćete grešku.

```
PC>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 30.0.0.1: Destination host unreachable.
Reply from 30.0.0.1: Destination host unreachable.
Reply from 30.0.0.1: Destination host unreachable.
Reply from 30.0.0.1: Destination host unreachable.

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss)
```

Ovaj primjer pokazuje kako kompanije koriste NAT za skrivanje njihove unutarnje mreže od vanjskog svijeta. Sada otvorite *web browser* sa bilo kojeg računala u 30.0.0.0 mreži i pretražite 50.0.0.1 mrežnu lokaciju.



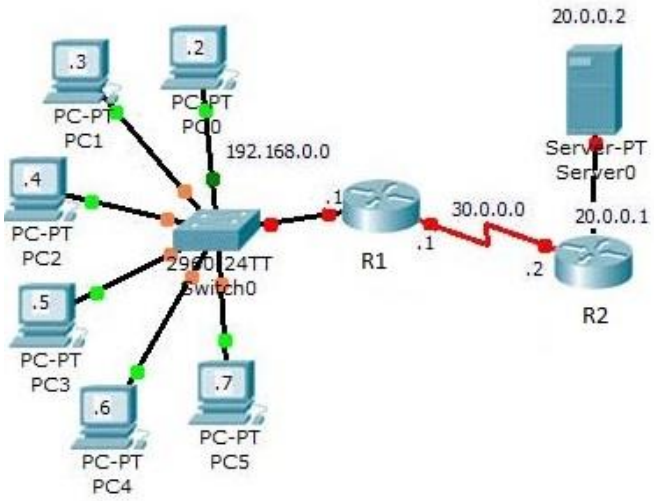
Slika 11.5. *Rješenje za NAT - zadatak 1*

Kao što vidite na slici, možete pristupiti sadržaju na 50.0.0.1

Zadatak - Konfiguracija dinamičkog NAT-a

Kod dinamičkog NAT-a, moramo manualno definirati dva seta adresa na uređaju za translaciju adresa. Prvi set definira kojim unutarnjim adresama je dozvoljeno da se translatiraju, a drugi set definira vanjske (globalne) adrese u koje će se unutarnje adrese translatirati.

U CPT-u nacrtajte topologiju kao na slici i dodjelite sučeljima IP adrese



Slika 11.6. *Mrežna topologija zadatka konfiguracije dinamičkog NAT-a*

U ovom primjeru naša unutarnja mreža je 192.168.0.0. Imamo 5 javnih IP adresa 50.0.0.1 do 50.0.0.5 za korištenje. **R1(1841 Router)** će biti NAT uređaj.

Konfigurirajte R1 po koracima navedenim ispod.

```
Router>enable
Router#configure terminal
Router(config)#hostname R1
R1(config)#interface fastethernet 0/0
R1(config-if)#ip address 192.168.0.1 255.0.0.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 30.0.0.1 255.0.0.0
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#ip  route  0.0.0.0  0.0.0.0  serial  0/0/0  (definira  statičko
usmjeravanje – „sve pakete koje neznamo gdje poslati, slati na serijski link
0/0/0“)
R1(config)#access-list 1 permit 192.168.0.0 0.0.0.255
        (definira list 1– listu unutarnjih adresa za translaciju)
R1(config)#ip  nat  pool  test  50.0.0.1  50.0.0.5  netmask  255.0.0.0
        (definira pool test– set globalnih adresa u koje će se translatirati
unutarnje adrese)
R1(config)#ip nat inside source list 1 pool test
(spaja list 1 i pool test)
R1(config)#interface fastEthernet 0/0
R1(config-if)#ip nat inside(definira da ovo sučelje spaja unutarnju mrežu)
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip nat outside (definira da ovo sučelje spajavanjsku mrežu)
R1(config-if)#exit
R1(config)#exit
```

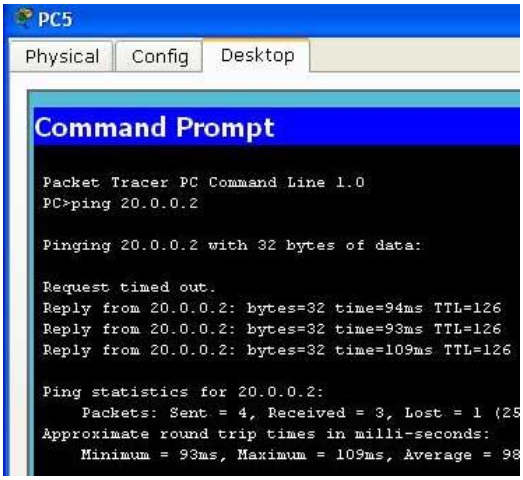
Konfiguracija usmjerivača R2

```
Router>enable
Router#configure terminal
Router(config)#interface fastEthernet 0/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#ip  route  0.0.0.0  0.0.0.0  serial  0/0/0  (definira  statičko
usmjeravanje – „sve pakete koje neznamo gdje poslati, slati na serijski link
0/0/0“)
Router(config)#hostname R2
```

Za testiranje NAT funkcionalnosti idi na R1 i omogući „debug“ za NAT iz „privilege“ moda

```
R1#debug ip nat
```

Sada idi na pc i pošalji naredbu ping na 20.0.0.2



Slika 11.7. *Prikaz ispisa „ping“ naredbe nakon konfiguracije dinamičkog NAT-a*

Kada ICMP ping paket stigne do R1, on provjerava izvorišnu adresu sa listom 1. Obzirom da je ovaj paket poslan iz mreže 192.168.0.0 on će zadovoljiti kriterije liste 1. Usmjerivač sada provjerava NAT pool kako bi pronašao slobodnu adresu za tranlaciju.

Proces možete vidjeti u izlazu debag naredbe na R1

```
IP NAT debugging is on
NAT: s=192.168.0.7->50.0.0.1, d=20.0.0.2[1]
NAT*: s=20.0.0.2, d=50.0.0.1->192.168.0.7[1]
NAT: s=192.168.0.7->50.0.0.1, d=20.0.0.2[1]
NAT*: s=20.0.0.2, d=50.0.0.1->192.168.0.7[1]
NAT: s=192.168.0.7->50.0.0.1, d=20.0.0.2[1]
NAT*: s=20.0.0.2, d=50.0.0.1->192.168.0.7[1]
NAT: s=192.168.0.7->50.0.0.1, d=20.0.0.2[1]
NAT*: s=20.0.0.2, d=50.0.0.1->192.168.0.7[1]
```

Kao što vidite u izlazu, izvorišna adresa 192.168.0.5 se translatirala u 50.0.0.1 prije nego je napustila usmjerivač. Sada provjerite web pristup sa bilo kojeg računala

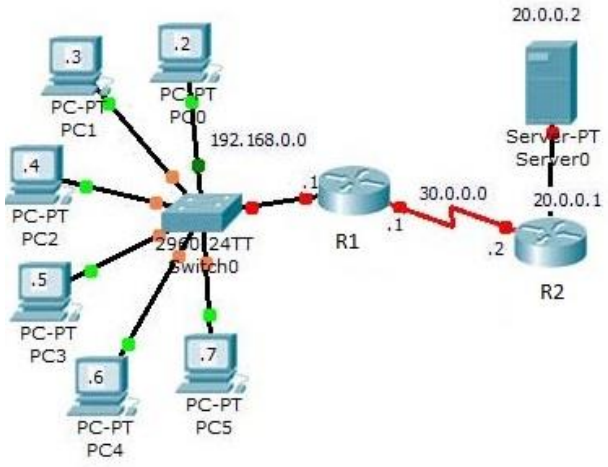
Slika 11.8. *Dohvat web stranice*

U stvarnom životu najbolje je isključiti „*debug*“ nakon testiranja i izaći iz *debug* moda.

```
R1#no debug ip nat IP NAT debugging is off R1#
```

Zadatak - Konfiguracija PAT-a

U dinamičkom NAT-uz translatiranje se radi iz IP adrese u IP adresu. Ukoliko želimo osigurati da svi unutarnji uređaji mogu istovremeno koristiti vanjske resurse, moramo imati jednak broj globalnih IP adresa na raspolaganju koliko imamo i unutarnjih. Ovo postaje problem ukoliko imamo na raspolaganju samo nekoliko globalnih adresa, a stotine unutarnjih koje treba translatirati. U takvim situacija moramo koristiti PAT. Za primjer ćemo koristiti istu topologiju koju smo imali kod dinamičkog NAT-a ali ovaj put ćemo koristiti samo jednu globalnu IP adresu 50.0.0.1



Slika 11.9. *Mrežna topologija zadatka konfiguracije PAT-a*

IP adrese na računalima su već konfigurirane pa treba samo konfigurirati R1 i R2

Konfiguracija R1

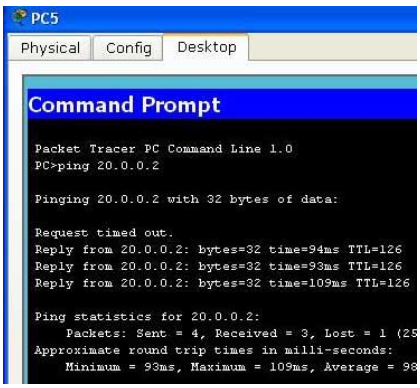
```
Router>enable
Router#configure terminal Enter configuration commands, one per line. End
with CNTL/Z.
Router(config)#hostname R1
R1(config)#interface fastEthernet 0/0
R1(config-if)#ip address 192.168.0.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 30.0.0.1 255.0.0.0
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#ip route 0.0.0.0 0.0.0.0 serial 0/0/0 (definira statičko
usmjeravanje – „sve pakete koje neznamo gdje poslati, slati na serijski link
0/0/0“)
R1(config)#access-list 1 permit 192.168.0.0 0.0.0.255 (definira list 1)
R1(config)#ip nat pool test 50.0.0.1 50.0.0.1 netmask 255.0.0.0 (definira
pool test koji u ovom slučaju ima samo jednu IP adresu)
R1(config)#ip nat inside source list 1 pool test overload (spaja list 1 i
pool test, te parametrom overload kaže da koristimo PAT)
R1(config)#interface fastEthernet 0/0
```

```
R1(config-if)#ip nat inside    (definira da je na ovo sučelje spojena „unutarnja" mreža)
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip nat outside   (definira da je na ovo sučelje spojena „vanjska" mreža)
R1(config-if)#exit
R1(config)#
```

Sada konfigurirajte R2

```
Router>enable
Router#configure terminal
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface fastEthernet 0/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#hostname R2
R2(config)#ip route 0.0.0.0 0.0.0.0 serial 0/0/0 (definira statičko usmjeravanje – „sve pakete koje neznamo gdje poslati, slati na serijski link 0/0/0")
```

Sada idite na bilo koji PC i pošaljite naredbu ping na adresu 20.0.0.2



Slika 11.10. *Prikaz ispisa „ping" naredbe nakon konfiguracije PAT-a*

Za provjeru rada PAT-a idite na R1 i pokrenite „show ip nat translations"

```
R1#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
icmp 50.0.0.1:1        192.168.0.7:1      20.0.0.2:1         20.0.0.2:1
icmp 50.0.0.1:2        192.168.0.7:2      20.0.0.2:2         20.0.0.2:2
icmp 50.0.0.1:3        192.168.0.7:3      20.0.0.2:3         20.0.0.2:3
icmp 50.0.0.1:4        192.168.0.7:4      20.0.0.2:4         20.0.0.2:4
```

Kao što vidite, ovaj put adresna translacija se radi putem port adresa, a ne putem IP adresa!

12.DHCP, DNS i E-mail usluge u mreži

DHCP (*Dynamic Host Configuration Protocol)*

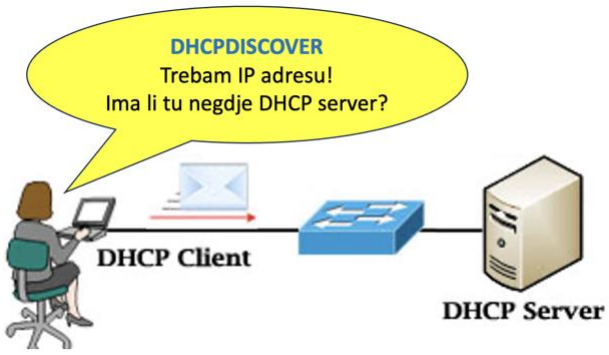
U IP okruženju, svako računalo treba imati svoju IP adresu kako bi mogao komunicirati sa drugim računalima. Postoje dva načina konfiguracije IP adresa:

- Statička (manualno postavimo IP adresu na računalo) i
- Dinamička (računalo po uključenju automatski dohvaća IP adresu pomoću DHCP protokola)

Velika prednost korištenja DHCP-a je u mogućnosti spajanja računala na mrežu bez da detaljno poznajemo mrežne konfiguracije. Npr. ukoliko idete u neki dućan, ili u kućni posjet prijatelju, poznavanjem wi-fi zaporke ili spajanjem kabelom na mrežu, vaše računalo odmah je spremno za rad. Da nema DHCP-a morali bi uvijek tražiti mrežnog administartora da nam da IP adresu koju možemo koristiti, te bi je svaki put morali manualno postavljati na računalo.

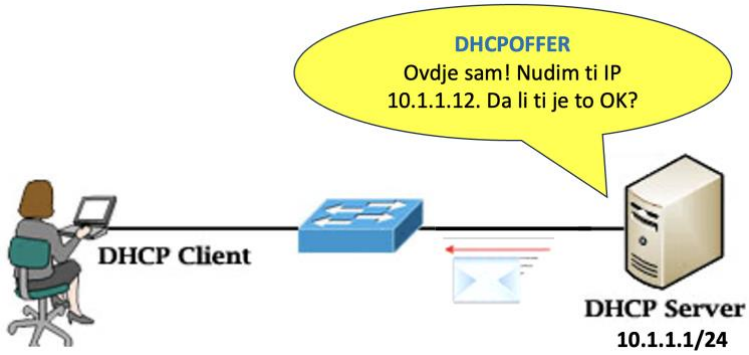
Na koji način nam DHCP pribavlja adrese?

1.Kada prvi put upalimo uređaj (računalo, tablet i sl.) ili kada se pokušavamo spojiti na mrežu, on mora dobiti IP adresu. Prva stvar koju uređaj radi je odašiljanje **DHCPDISCOVER**poruke na lokalnu podmrežu. Obzirom da uređaj (client) nema nikakvog saznanja u kojoj podmreži se nalazi, tu poruku šalje na *broadcast* adresu svih podmreža (odredišna IP adresa 255.255.255.255, koja je *layer 3 broadcast address*) i odredišnu MAC adresa FF-FF-FF-FF-FF-FF (koja je *layer 2 broadcast address*). Uređaj još nema svoju IP adresu, pa za adresu izvora poruke postavlja IP adresu 0.0.0.0. Svrha slanja DHCPDISCOVER poruke je pokušaj pronalaženja DHCP Servera koji nam može dodjeliti IP adresu.



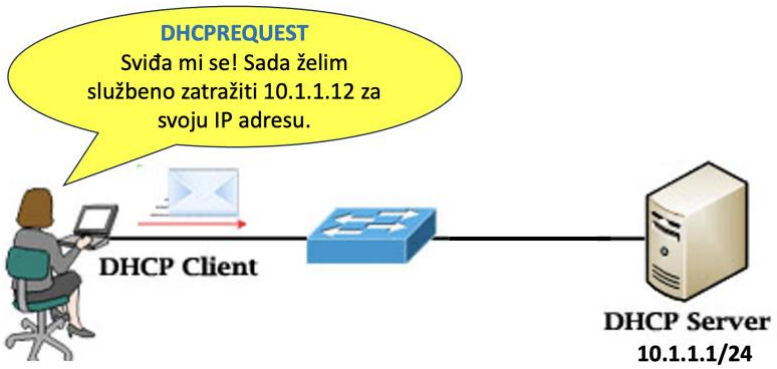
***Slika 12.1.** Naredba „DHCPDISCOVER“ [6]*

2.Nakon primanja DHCPDISCOVER poruke , DHCP Server će dinamički dohvatiti neku nedodjeljenu IP adresu iz postavljenog skupa IP adresa za korištenje u toj podmreži(*IP address pool*)i broadcast porukom **DHCPOFFER**poslati će je nazad klijentu. DHCPOFFER poruka može sadržavati i dodatne informacije poput maske podmreže, default gateway,vremena trajanja adrese, i adresu DNS servera.



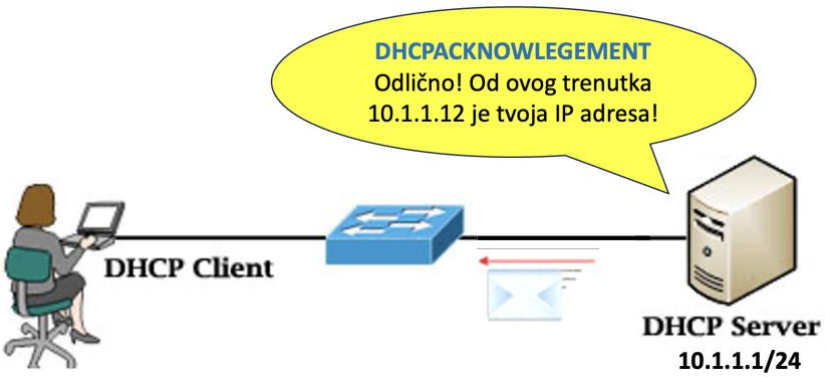
Slika 12.2. *Naredba „DHCPOFFER“ [6]*

3.Ukoliko klijent (naš uređaj) prihvati ponuđenu adresu, on šalje broadcast poruku **DHCPREQUEST**kojom potvrđuje zauzimanje te IP adrese. U nazivu ove poruke još stoji dio „request“ jer uređaj ne mora nužno prihvatiti ponuđenu IP adresu već može od servera zatražiti novu. DHCPREQUEST je još uvijek broadcast poruka jer klijent (tj. naš uređaj) još nije primio potvrdu dodjele IP darese od strane servera.



Slika 12.3. *Naredba „DHCPREQUEST“ [6]*

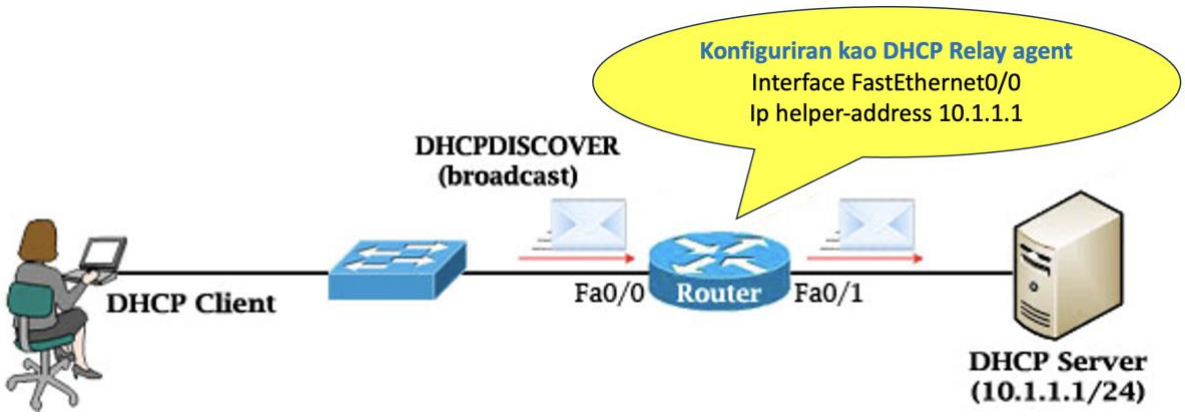
4.Kada DHCP Server primi DHCPREQUEST poruku od klijenta, server prihvaća zahtjev slanjem unicast**DHCPACKNOWLEDGEMENT** (DHCPACK) poruke klijentu.



Slika 12.4. *Naredba „DHCPACKNOWLEGMENT“ [6]*

Po primitku DHCPACKNOWLEDGEMENT poruke, IP adresa se smatra konačno dodjeljena klijentu. Klijent uobičajeno zadržava dodjeljenu adresu uz povremeno javljanje DHCP serveru kako bi se obnovilo trajanje te dodjeljene adrese.

Ukoliko DHCP server nije na istoj podmreži kao i klijent, moramo konfigurirati usmjerivač na strani klijenta koji će raditi kao DHCP Relay Agent, tj. da propušta DHCP porukle između klijenta i servera. Kako bi to napravili moramo samo na usmjerivaču postaviti "ip helper-address <*IP-address-of-DHCP-Server*>" naredbu na sučelje koje prima DHCP poruke od klijenta.



Slika 12.5. *DHCP Relay agent [6]*

Kao što znamo, usmjerivači ne prenose broadcast pakete (već ih odbacuju) pa bi u tom slučaju DHCP poruke poput DHCPDISCOVER bile odbačene. Postavljanjem "ip helper-address …" naredbe, usmjerivač će prihvatiti broadcast poruku i pretvoriti je u unicast poruku te je proslijediti na adresu DHCP servera. Odredišna IP adresa uzima se iz postavljene "ip helper-address …" naredbe.

Kada se događa DHCP konflikt adresa

Tijekom procesa dodjele IP adrese, DHCP server provjerava dostupnost adrese slanjem ping naredbe na tu adresu, prije nego je dodjeli klijentu. Ukoliko nema odgovora na ping naredbu, DHCP server smatra da je IP adresa slobodna i da je može dodjeliti klijentu. Ukoliko neko odgovori na ping naredbu, DHCP server zabilježi konflikt, tu adresu miče iz skupa slobodnih adresa i neće je nikome dodjeljivati sve dok mrežni administrator manualno ne riješi problem.

Algoritmi dodjela IP adresa

DHCP je protokol tipa klijent-server, koji omogućava automatsku dodjelu IP adresa uređajima. Postoje tri metode dodjele IP adresa:

— **Dinamičko dodjeljivanje IP adresa**. Administrator mreže dodjeljuje opseg IP adresa DHCP serveru. Svaki klijent u LAN-u konfiguriran je tako da traži IP adresu od servera, u fazi inicijalizacije. Taj proces "zahtijev — odgovor" funkcionira po principu dodjeljivanja adresa na određeno vreme. Poslije isteka tog vremena, vrši se obnavljanje. Klijent zadržava adresu za slijedeći period ili mu se dodjeljuje druga IP adresa.

— **Automatsko dodjeljivanje IP adresa**. Postupak je sličan dinamičkom dodjeljivanju adresa, s tom razlikom što DHCP server održava tablicu dodjeljenih IP adresa. Kada istekne vremenski period, prvi izbor DHCP servera je da klijentu ponovo dodijeli istu IP adresu.

— Statičko dodjeljivanje IP adresa. DHCP održava tablicu sa parovima IP adresa/MAC adresa. Tu tablicu manualno popunjava administrator mreže. IP adrese se dodjeljuju samo registriranim klijentima, odnosno samo klijentima čije se MAC adrese nalaze u spomenutoj tabeli.

Postavljanje DHCP servisa

DHCP servis prije se isključivo pokretao sa servera dok se danas može pokretati i na drugim mrežnim elementima npr. na usmjerivačima.

Ukoliko želimo pokrenuti DHCP servis na Cisco usmjerivačima koristimo slijedeće naredbe:

Konfiguracijska naredba	**Opis**
Router(config)#ip dhcp pool CLIENTS	Kreira DHCP skup s nazivom „CLIENTS“
Router(dhcp-config)#network 10.1.1.0 /24	Određuje mrežu i mrežnu masku DHCP adresnog skupa
Router(dhcp-config)#default-router 10.1.1.1	Postavlja *gateway* adresu za DHCP klijente
Router(dhcp-config)#dns-server 10.1.1.1	Postavlja adresu DNS poslužitelja za DHCP klijente
Router(dhcp-config)#domain-name 9tut.com	Postavlja ime domene
Router(dhcp-config)#lease 0 12	Određuje vrijeme trajanja adrese. Sintaksa je “**lease** {days[hours] [minutes] | infinite}”. U ovom slučaju vrijeme trajanja je 12 sati.
Router(dhcp-config)#exit	
Router(config)# ip dhcp excluded-address 10.1.1.1 10.1.1.10	IP adresni skup koji DHCP poslužitelj neće dodjeljivati klijentima. Pazi – ova naredba se postavlja iz globalnog konfiguracijskog moda.

Tablica 12.1. Naredbe za konfiguraciju DHCP servisa

DNS (*Domain Name System*)

Adresiranje u računalnim mrežama vrši se pomoću računalnih adresa. U danas najvećoj računalnoj mreži adresiranje nje se vrši IP adresama koje su brojčane. Brojčane adrese su ljudima teško pamtljive i ne možemo ih asociativno povezivati sa nekim pojmovima. Zato se kao imena domena koriste riječi koje nama imaju određeno značenje. *Domain name system*, poznatiji kao "DNS" je mrežni sustav koji nam omogućuje povezivanje nama lako pamtljivih imena domena sa njihovim mrežnim adresama.

Ime domene (*Domain Name*)

Ime domene je ljudima prilagođen naziv koji se koristi za opisivanje nekog internetskog resursa. Npr. "google.com" je ime domene. URL "google.com" je povezan sa mrežnim serverima u vlasništvu firme Google Inc.

IP Adresa

IP adresa je mrežna adresa određenog resursa. Svaka javna IP adresa mora biti jedinstvena. IPv4 je najčešće upotrebljavan sustav adresiranja. DNS sustav nam spaja imena domena sa mrežnim adresama kako mi ne bi morali pamtiti komplicirane nizove brojeva za svaku mrežnu lokaciju kojoj želimo pristupiti preko mreže (Interneta).

Top-Level Domain

Top-level domain, ili TLD, je najopćenitiji dio domene. To je onaj dio naziva imena domene koji se nalazi krajnje desno (desno od točke). Najčešće TLD su "com", "net", "org", "gov", "edu", i "io". Top-level domene su na vrhu hijerarhije sustava domena. Na svijetu postoji određen broj organizacija kojima je ICANN (Internet Corporation for Assigned Names and Numbers) dala dopuštenje da imaju upravljačku kontrolu nad tim domenama. Te organizacije vrše raspodjelu imena domena pod određenom TLD i formiraju registar domena kako bi se osigurala jednoznačnost (tj. da je svako dodjeljeno ime jedinstveno na svijetu)

Uređaji i poddomene (*Hosts & SubDomains*)

Unutar neke dodjeljene domene („ljudsko“ ime za brojčanu IPv4 adresu), vlasnik domene može postaviti jedan ili više uređaja koji pružaju različite usluge (servise). Npr. možete imati web-server koji će biti dostupan pod čistim imenom domene (example.com) i također kroz tzv. "host" definiciju "www" (www.example.com).

Može se definirati pristup i ostalim uslugama poput ftp-a definiranjem uređaja koji se zove "ftp" ili "files" (ftp.example.com ili files.example.com).

Domenska imena uređaja mogu biti proizvoljno duga dokle god su jedinsvena unutar domene. Domenski nazivi za imena „uređaja“ nazivaju se poddomene (*SubDomains).*

Poddomena se odnosi na domenu koja je dio veće domene. Pa tako pod TLD domenom **.com** imamo i *google.com* i *ubuntu.com.* U ovom slučaju možemo reći da je *google.com* poddomena od domene *.com.*

Istom analogijom možemo imati više podjela poput www.oss.unist.hr gdje je „oss“ (odjel za stručne studije) poddomena od Sveučilišta u Splitu „unist.hr“. Razlika između imena uređaja (*hosts*) i imena podomene (*subdomains*) je u tome što podomena služi za širenje definicije same domene.

Bez obzira da li govorimo o imenu uređaja ili poddomene, lijevi dio naziva je specifičniji, a segmenti koji idu prema desnoj strani sve su općenitiji.

Name Server

Name server je računalo koje je zaduženo za prevođenje imena domena u brojčane IP adrese. Ovi serveri odrađuju većinu posla u DNS sustavu. Obzirom da je totalni broj translacija koji se obavlja svake sekunde u svijetu prevelik za bilo koji server, svaki od tih servera može preusmjeriti zahtjeve ka drugim *name serverima* ili razdijeliti odgovornosti prema grupama podomena. Znači oni mogu ili odmah odgovoriti na zahtjev za translacijom (prevođenjem) ili mogu proslijediti upit drugom serveru.

Zone File

Zone file je jednostavni tekstualni zapis koji sadrži veze između imena domene i njene IP adrese. Pomoću njega DNS sustav pronalazi IP adresu tražene domene.

Zapisi (*Records*)

Zone file sastoji se od pojedinačnih zapisa. Zapis je najjednostavnija tekstualna forma povezuje ime domene sa njenom IP adresom. Postoji više vrsta zapisa i sada ćemo proći samo kroz one najvažnije

SOA Records

The Start of Authority, ili SOA, zapis je obvezan u svim zone file-ovima. Mora biti prvi pravi zapis koji će se postaviti u tom file-u i najkopliciraniji je za razumjevanje. U ovom kolegiju nećemo ići u detalje već ćemo objasniti samo dio neophodan za aktivaciju DNS servisa u CPT programu.

ns1.domain.com.: Definira primarni *master name server* za ovu domenu. *Name server* mogu biti ili *master* ili *slaves*, i ukoliko konfiguriramo dinamički DNS barem jedan mora biti konfiguriran kao "primary master", što upisujemo ovdje.

A i AAAA zapis

Povezuje uređaj sa IP adresom. "A" zapis se koristi za povezivanje uređaja sa IPv4 adresom, a "AAAA" zapis za povezivanje uređaja i IPv6 adresom.

Format zapisa izgleda:

```
host     IN      A       IPv4_address
host     IN      AAAA    IPv6_address
```

CNAME zapis

CNAME zapis definira alias nazive vašeg servera koji je već definiran A ili AAAA zapisom. Pojednostavljeno, ovdje upisujemo sve varijacije imena za koje mislimo da bi se mogle pojaviti. Npr. ako smo u glavnom tj. A zapisu naš uređaj nazvali "server1", sada ovdje možemo staviti "www" kao alias za ovaj uređaj:

```
server1     IN  A       111.111.111.111
www         IN  CNAME   server1
```

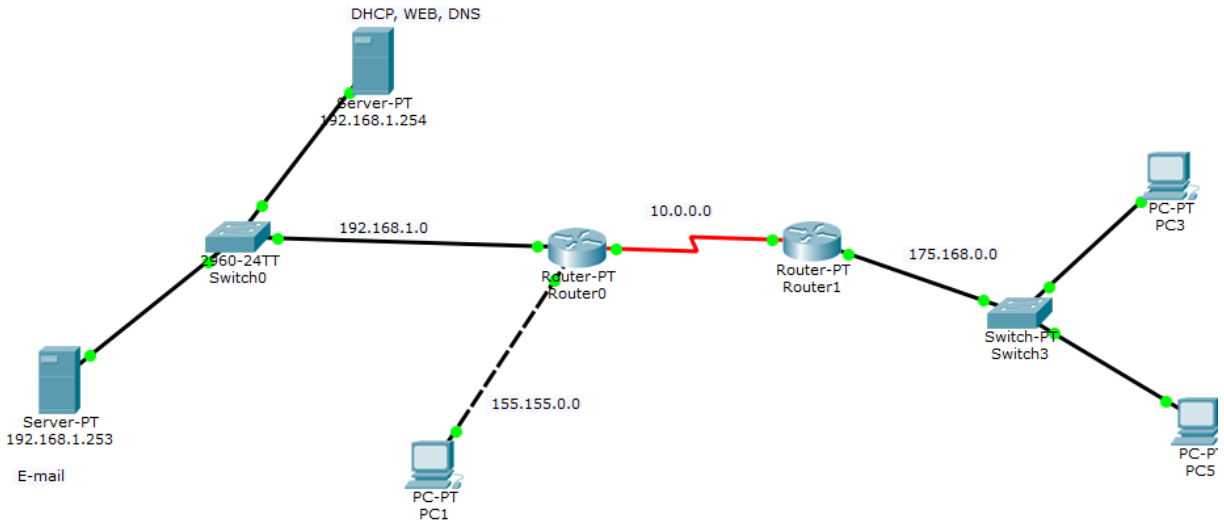
Ovo radimo jer će neko na web pregledniku postaviti adresu „server1“, a neko će pak postaviti www.server1 za pristup ovom uređaju.

Postavljanje web poslužitelja

U ovom kolegiju nećemo se baviti detaljima problematike web stranica. Kod servera u CPT - web server funkcionalnost aktivira se jednostavnim uključenjem servisa, a na samom serveru već su postavljane određene test stranice.

Zadatak – Aktivacija i konfiguracija DHCP, DNS i E-mail usluga

U CPT-u nacrtajte mrežnu topologiju kao na slici, kreirajte sučelja i dodijelite IP adrese po pravilima, te postaviti statičko usmjeravanje po „last resort“ principu.

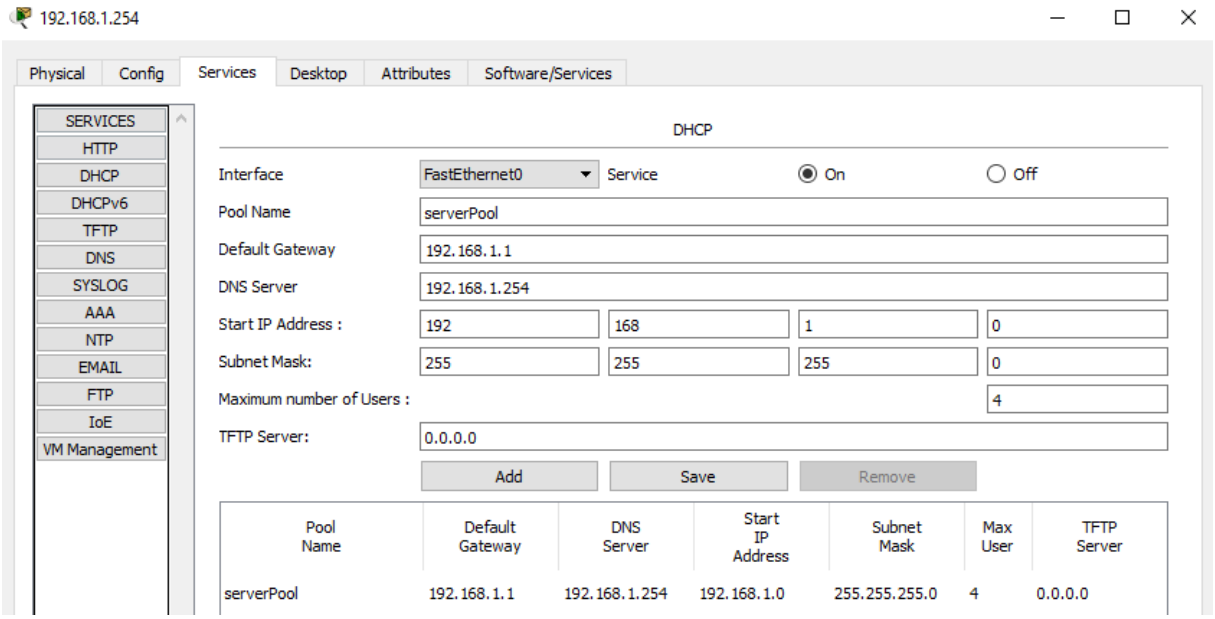


Slika 12.6. *Mrežna topologija za zadatak*

Web, DNS i DHCP serveru dodjelite statičku IP adresu 192.168.1.254

E-mail serveru dodijelite statičku IP adresu 192.168.1.253

Aktivirajte DHCP servis (test) sa samo 4 člana



Slika 12.7. *Pokretanje DHCP servisa kroz GUI*

Sada dodajte još 3 računala na preklopnik u 192.168.1.0 mreži, te postavite DHCP opciju za konfiguraciju IP adrese. Što primjećujete?

Sada dodajte još jedno računalo. Što primjećujete?

Veza ima zelenu indikaciju, ali kada idete provjeriti IP adresu računala, DNS server i gateway,...vidite da ih nema. Ukoliko pokušate „pingati“ sa tog računala bilo koje drugo, vidite da veza ne funkcionira. To je zato jer su sve adrese već iskorištene! 3 računala i adresa sučelja usmjerivača koja je već prije postavljena. Ukoliko želimo priključiti još računala, moramo povećati skup članova u DHCP-u ili ostalim računalima manualno upisivati IP adrese.

Aktivacija DNS servisa

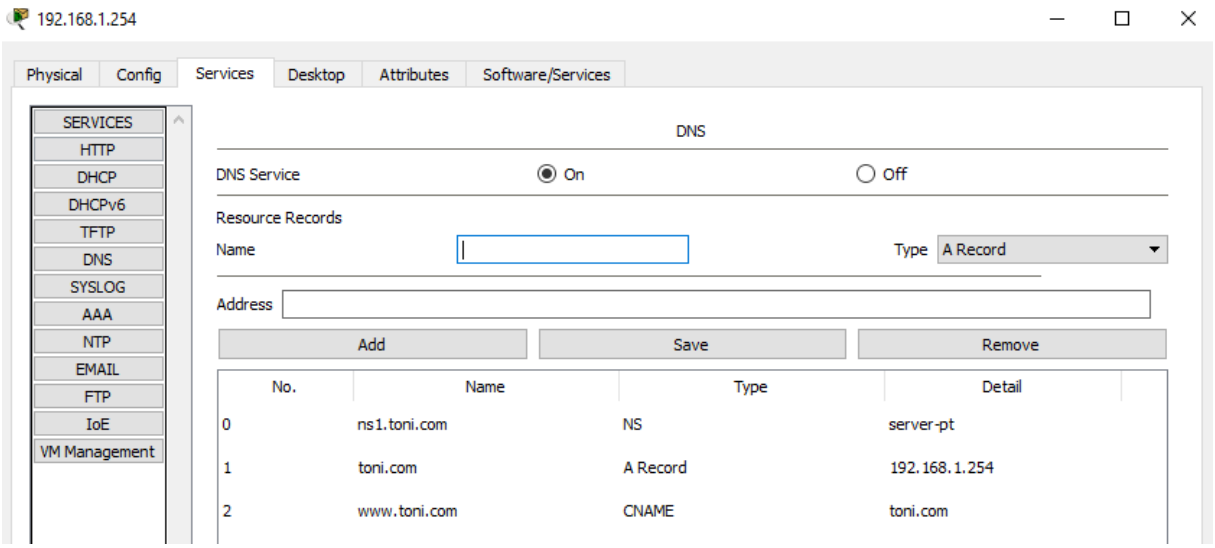
Na serveru se već nalaze određene web stranice. Mi sada u DNS servisu možemo definirati „ljudsko ime“ tog web sadržaja, npr. toni.com

Kao što smo rekli, za svaku web domenu (ljudsko ime), u DNS-u moramo prvo definirati „pravo ime“ domene, tzv. A Record. (toni.com)

Nakon toga moramo definirati ns1 zapis koji nas vodi na glavni poslužitelj za tu domenu

Nakon toga definiramo sve varijante imena koje bi se mogle koristiti - CNAME

Na poslužitelju otiđite na DNS servis i izvršite konfiguraciju kao na slici



Slika 12.8. *Pokretanje DNS servisa kroz GUI*

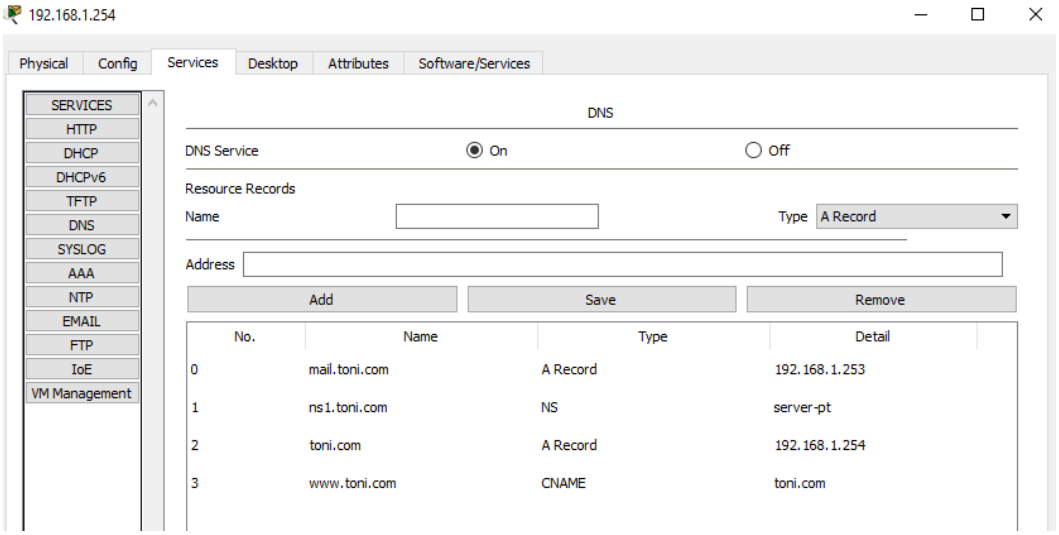
Sada sa nekog računala iz web sučelja pokušajte pristupiti sadržaju na toni.com.

Sada pokušajte sa www.toni.com

Kao što vidite, radi u oba slučaja.

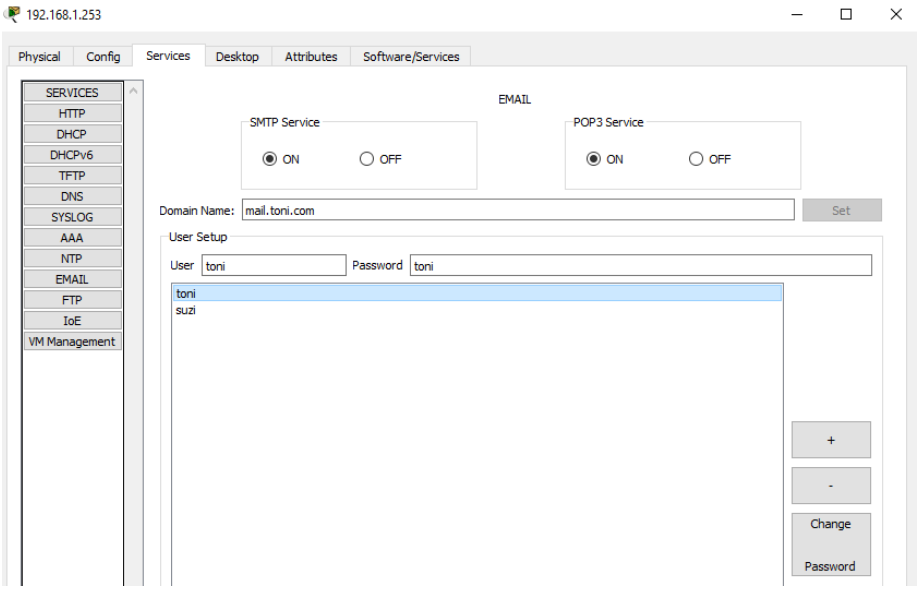
E-mail usluga

Ukoliko želimo pokrenuti E.-mail servis, također moramo napraviti zapis u DNS serveru. Recimo da želimo konfigurirati set adresa na „mail.toni.com“



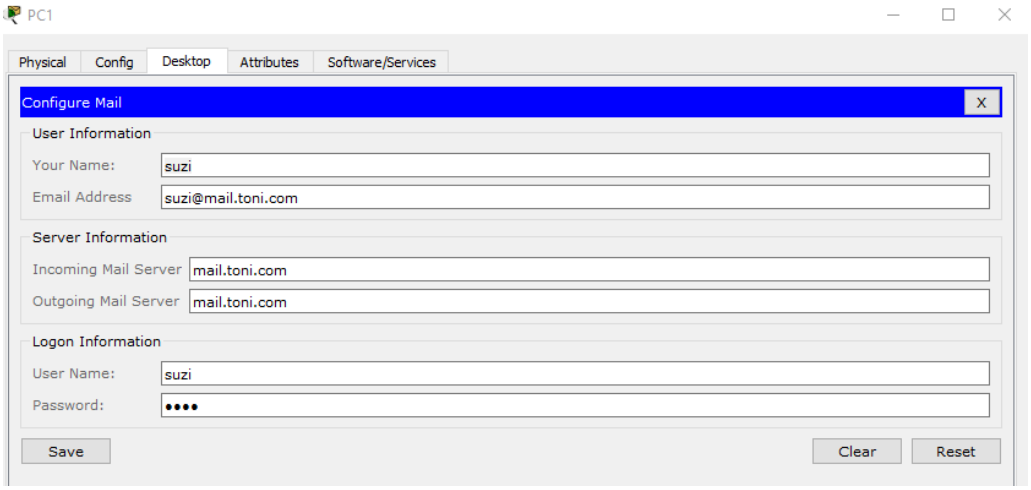
Slika 12.9. *Povezivanje E-mail usluge sa DNS-om*

Kada smo napravili taj zapis u DNS servisu, možemo ići na E-mail poslužitelj i postaviti adrese. Kreirajmo dva korisnika toni i suzi (adrese toni@mail.toni.com i suzi@toni.com)

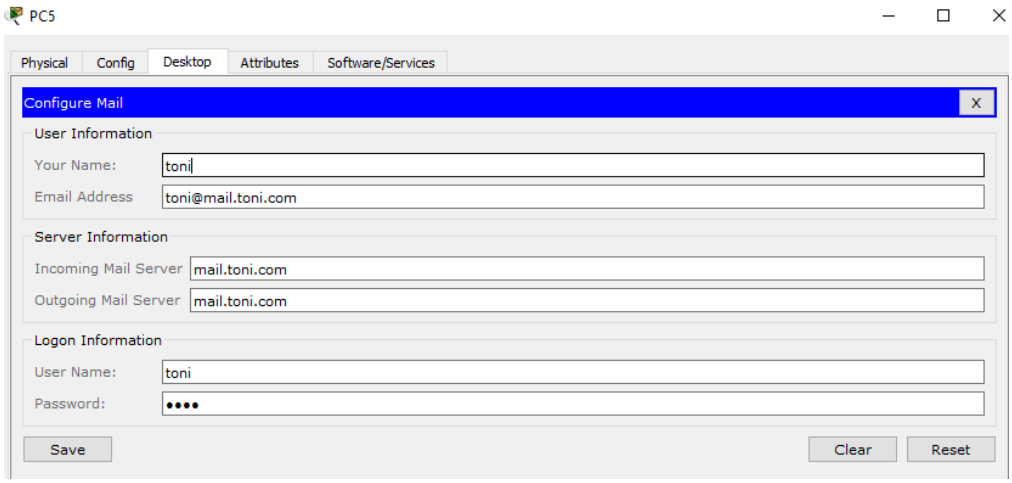


Slika 12.10. *Kreiranje korisnika na E-mail poslužitelju*

Nakon toga moramo ići na računalo, te tamo također kreirati E-mail. Na jednom računalu ćemo kreirati račun za korisnika „suzi“, a na jednom za korisnika „toni“



Slika 12.11. *Kreiranje E-mail korisnika „suzi“ na računalu PC1*



Slika 12.12. *Kreiranje E-mail korisnika „toni“ na računalu PC5*

Sada probajte poslati E-mail sa jednog na drugo računalo.

13. Pristupne ili „*Access control*“ liste

Što su to *Access Control Lists* (ACL)?

ACL određuje koji korisnici ili procesi imaju dozvolu pristupa nekom „objektu“, te također definiraju i dozvoljene radnje. Svaki unos u ACL uobičajeno određuje korisnika i operaciju. Na primjer, ukoliko neki objekt (npr. direktorij ili datoteka u bazi podataka) posjeduje ACL koji sadrži (Sanja: read,write; Mate: read), ovo će dozvoliti Sanji pregled (čitanje) i pisanje u nekom objektu, dok će Mate moći samo pregledati objekt (datoteku).

Generalno, uporabu ACL-a možemo podijeliti po namjeni na:

1. ACL za pristup različitim datotekama ili bazama podataka
2. Mrežne ACL

ACL za pristup različitim datotekama ili bazama podataka (*filesystem ACL*)

Ovakve ACL uobičajeno su napravljene u obliku tablice gdje unosi određuju ovlasti pojedinačnog korisnika ili pak grupe korisnika koje će se primjeniti na neki određeni „objekt“ koji može biti program, proces ili neka baza podataka. Unosi u tablicu nazivaju se *access-control entries* (ACEs) u Microsoft Windows NT, OpenVMS, Unix, i Mac OS X operativnim sustavima. Svaki „object“ kojeg korisnik može „dohvatiti“ ima identifikator svoje ACL. Prava ili dozvole određuju specifična prava pristupa tj. određuju može li korisnika samo čitati, pisati ili pak pokrenuti „objekt“ (jer „objekt“ može biti datoteka, ali i program).

Mrežne ACL

Neki HW elementi računalnih mreža (posebno usmjerivači i preklopnici) također imaju svoje ACL liste. One najčešće služe za određivanje pravila koja određuju brojeve portova ili pak IP adrese koje se mogu koristiti na nekom uređaju, a preko toga se nekim korisnicima dozvoljava ili ne korištenje određenih usluga. Iako se ACL mogu konfigurirati i korištenjem mrežnih naziva uređaja (*network domain names*), to nije preporučljivo jer se u TCP, UDP i ICMP zaglavlja ne prenose domenska imena. To bi kao posljedicu imalo uvođenje dodatnog procesa koji bi morao povezivati domenska imena sa IP adresama. Ovakvi procesi predstavljaju dodatni sigurnosni rizik za sustave koje ACL štite. Individualni računalni serveri kao i usmjerivači mogu imati svoje mrežne ACL liste. *Access control* liste mogu biti konfigurirane da kontroliraju ulazni i izlazni promet pa u tom kontekstu imaju funkciju sličnu vatrozidu (*firewall*). Poput vatrozida, ACL također podliježu sigurnosnoj regulaciji i standardizaciji poput PCI DSS.

Obzirom da ovu vježbu izvodimo u skolopu kolegija „Širokopojasnih mreža“, obrađivati ćemo samo problematiku mrežnih ACL-a. Također, obzirom da praktični dio nastave izvodimo na CPT-u, fokusirati ćemo se na naredbe koje se koriste u Cisco operativnim sustavima.

Pregled mrežnih *Access* Lista

Access liste se koriste za izradu vatrozid zaštite na usmjerivačima. *Access* liste su postavke koje usmjerivač primjenjuje kod prosljeđivanja prometa. Ukoliko pronađe podudarnost sa postavkama iz ACL, usmjerivač filtrira promet paketa na način da ga ili propušta ili zabranjuje. Kod Cisco operativnog sustava postoje 3 osnovne vrste ACL-a.

- **Standardna *access* lista** (označavamo ih brojevima od 1 do 99 i 2000 do 2699),
- **Proširena (*extended*) *access* lista**
- ***Access* lista sa nazivom.** (Lista je označena imenom umjesto sa brojem, a može biti konfigurirana ili kao standardna ili proširena *access* lista)

Način rada

- ACL se pišu i izvršavaju liniju po liniju!
- Svaki unos u ACL je zasebna izjava ili pravilo!
- ACL prestaje s radom kada se pronađe prva podudarnost!
- Na kraju svake ACL postoji implicitna naredba “deny all” ili “deny any” koja nije upisana (ne vidite je) ali se podrazumjeva. Ovo često izaziva probleme jer ljudi krivo pretpostave da su ACL po „*defaultu*“ propusne, tj. da mi trebamo upisati samo one naredbe koje trebaju nekome zabraniti pristup kako bi filtrirali promet, a da će sav ostali promet proći kroz ACL. To je krivo!

Postupak izrade i pokretanje rada

Imamo dva koraka:

1. izrada *access* lista (standardne ili proširene)
2. primjena *access* liste na neko sučelje (za ulazni ili izlazni promet)

1. Izrada ACL

Standardne ACL (označavaju se brojevima 1-99 i 2000-2699)

Zabranjuju ili dozvoljavaju definirane:

1) izvorišne IP adrese

Proširene (*Extended*) ACL (označavaju se brojevima 100-199)

Zabranjuju ili dozvoljavaju definirane:

1) izvorišne IP adrese,

2) odredišne IP adrese,

3) port-ove (tj. aplikacije)

2. Primjena ACL

Gdje primjeniti ACL?

Standardnu ACL primjenjuje se za filtriranje ulaznog ili izlaznog prometa usmjerivačkog sučelja koje je najbliže prometnom odredištu.

Proširenu ACL primjenjuje se za filtriranje ulaznog ili izlaznog prometa usmjerivačkog sučelja koje je najbliže izvoru prometa.

Cisco IOS CLI naredbe za standarde ACL

Format naredba za kreiranje standardne access liste:

access-list <1-99><deny | permit><source ip address><wildcard bits>

ili

access-list <1-99><deny | permit> host <source ip address>

Primjeri:

Zabrani ili dozvoli mrežu klase c:

```
router(config)#access-list 1 deny 192.168.1.0 0.0.0.255 – kreira access listu broj 1 koja kaže da niti jedan uređaj iz ove mreže ne može proći (jer je wild-card označio sve uređaje)
router(config)#access-list 1 permit 192.168.2.0 0.0.0.255 – svi uređaji iz ove mreže prolaze
```

Zabrani ili dozvoli neki zasebni uređaj:

```
router(config)#access-list 1 deny 192.168.1.100 0.0.0.0 – prvi način zapisa naredbe
router(config)#access-list 1 deny host 192.168.1.100 – drugi način zapisa gornje naredbe


router(config)#access-list 1 permit 192.168.1.101 0.0.0.0 – prvi način zapisa naredbe
router(config)#access-list 1 permit host 192.168.1.101 – drugi način zapisa gornje naredbe
```

Zabrani ili dozvoli sve uređaje:

```
router(config)#access-list 1 deny any
router(config)#access-list 1 permit any
```

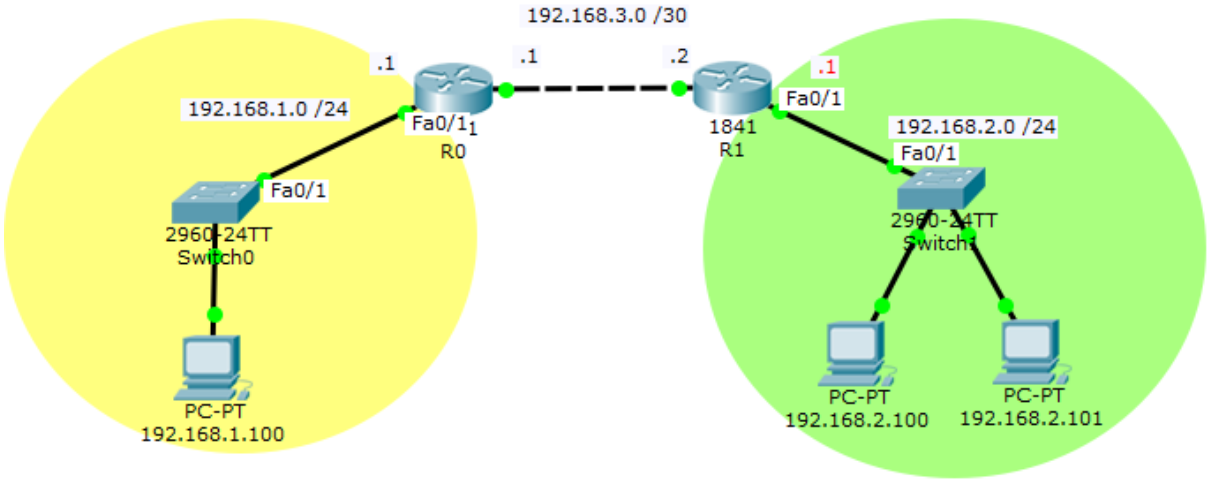
Primjeni ACL na neko usmjerivačko sučelje za ulazni (u usmjerivač) ili izlazni promet:

```
router(config)#interface fastethernet 0/0
router(config-if)#ip access-group 1 out – „1“ označava broj ACL-a koju smo prethodno kreirali.
router(config)#interface fastethernet 0/1
router(config-if)#ip access-group 1 in
```

Zadatak – Izrada standardne ACL

Pokrenite „acl-standard1-begin“ file za CPT koji se nalazi na Moodle-u. Statičke rute koje omogućuju promet između mreža već su konfigurirane, kao i sva sučelja

```
R0(config)#ip route 192.168.2.0 255.255.255.0 192.168.3.2
R1(config)#ip route 192.168.1.0 255.255.255.0 192.168.3.1
```



Slika 13.1. *Mrežna topologija za ACL zadatak*

„Zelena“ mreža predstavlja izvor prometa, tj. računala iz te mreže će pokušati „pingati“ računalo u „žutoj“ mreži koja predstavlja odredišnu mrežu.

Usmjerivačko sučelje „najbliže“ odredištu je FA0/1 na usmjerivaču R0

Usmjerivačko sučelje „najbliže“ izvoru je FA0/1 na usmjerivaču R1

Sva računala trenutno mogu komunicirati (provjeri naredbom „ping“)

a) Prvi zadatak u ovoj vježbi – primjeni **standardnu** ACL kako bi onemogućili samo računalu .2.101 iz zelene mreže da komuniciraju sa žutom

Obzirom da koristimo standardnu ACL – ona se uvijek primjenjuje na sučelju najbliže odredištu. Sučelje najbliže odredištu je FA0/1 na usmjerivaču R0. Sada znamo gdje trebamo kreirati i primjeniti ACL. Kreiramo je na usmjerivaču R0, a primjenjujemo na njegovom sučelju FA0/1. Želimo zaustaviti promet koji ide „prema“ žutoj mreži, a to je promet koji „izlazi“ iz usmjerivača kroz sučelje FA0/. Znači, blokirati ćemo izlazni (OUT) promet.

Prvo kreiramo ACL slijedećim naredbama:

`R0(config)#access-list 1 deny 192.168.2.101 0.0.0.0` – ova izjava (naredba) kaže da *access* lista „1“ zabranjuje prolaz paketima računala 192.168.2.101

Umjesto gornje naredbe mogli smo koristiti i drugu verziju te naredbe
`R0(config)#access-list 1 deny host 192.168.2.101).`
Da smo na primjer htjeli blokirati cijelu zelenu mrežu tada bismo koristili naredbu
`R0(config)#access-list 1 deny 192.168.2.100 0.0.0.255`

`R0(config)#access-list 1 permit any` – ova izjava (naredba) kaže da access lista „1“ propušta sve
Zašto nam treba ova linija naredbe? Zato što se svaki paket koji dođe na usmjerivač komparira

kroz ACL i to liniju po liniju dok se ne nađe podudarnost! Kada se podudarnost pronađe – ACL prestaje sa radom. Na kraju svake ACL postoji implicitna naredba "*deny all*" ili "*deny any*" koja nije upisana (ne vidite je) ali se podrazumjeva, pa kada ne bi bilo ove linije koda, svi paketi bili bi odbačeni.

Sada ćemo primjeniti našu ACL listu „1" na sučelje

```
R0(config)#int fa0/1
R0(config-if)#ip access-group 1 out
```

Sad ponovno provjerite međusobnu komunikaciju računala i vidjeti ćete da računalo 2.100 ima vezu, a računalo 2.101 nema vezu sa žutom mrežom.

Naredbom „*show run*" iz *privileged* moda provjerite trenutnu konfiguraciju usmjerivača i vidjeti ćete da je ACL 1 primjenjena na sučelje FA0/1

Ukoliko bi se standardna lista primjenjivala bliže izvoru, ona bi također blokirala promet određenog računala, ali NE SAMO prema našoj mreži, već i prema svim drugim mrežama!! Računalo bi u potpunosti izgubilo mogućnost komunikacije sa ostalim mrežama.

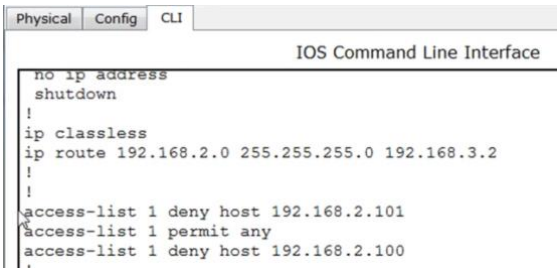
b) Drugi zadatak u ovoj vježbi – primjeni **standardnu** ACL kako bi onemogućili i drugom računalu (.2.100) iz zelene mreže da komuniciraju sa žutom

Prva pomisao koja nam pada na pamet je da samo odemo na R0 i da dodamo liniju koda

```
R0(config)#access-list 1 deny host 192.168.2.100
```

koja će nadograditi našu ACL 1.....međutim to neće raditi!

Uporabom „*show run*" naredbe pogledajmo kako bi u tom slučaju izgledala naša ACL 1 lista



Slika 13.2. *Prikaz rada ACL 1 liste po koracima*

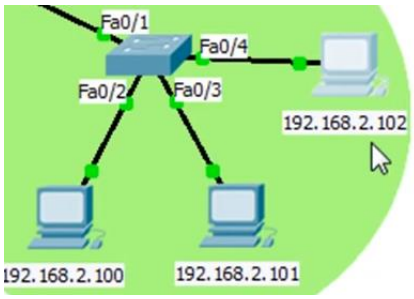
Vidimo da se između dvije zabrane nalazi „*permit any*". Kada dođe paket od računala 2.100 lista ga prvo komparira sa prvom izjavom i vidi da nema podudarnosti. Zatim ga komparira sa drugom izjavom i tu dolazi do podudarnosti, te se naredba izvršava – paket se propušta!!.., a lista završava sa radom i nikad se ne uvidi da je slijedeća izjava zapravo zabranjivala tom paketu prolaz u „žutu" mrežu.

Kako riješiti ovaj problem? – Moramo izbrisati našu ACL 1 i ponovno je cijelu kreirati!!

```
R0(config)#no access-list 1  – briše postojeću ACL listu
R0(config)#access-list 1 deny host 192.168.2.100
```

```
R0(config)#access-list 1 deny host 192.168.2.101
R0(config)#access-list 1 permit any
```

Dodajmo sada još jedno računalo u zelenu mrežu kako bismo se uvjerili da sve radi ispravno, tj. ovo treće računalo trebalo bi moći komunicirati sa žutom mrežom.



Slika 13.3. *Mrežna topologija uz dodatno računalo*

Nakon konfiguracije, provjerite komunikaciju „ping“ naredbom.

Proširene (extended) ACL

Proširene *access* liste nude nam veću fleksibilnost u odnosu na standardne jer nam omogućuju filtriranje prometa ne samo po izvorišnoj IP adresi, već i po odredišnoj IP adresi kao i po broju protokolnog porta tj. vrsti usluge (npr. TCP port 23 – telnet). Obzirom da sada možemo kao filter parametar koristiti izvorišnu ali i odredišnu IP adresu, možemo proširenu ACL listu primjenjivati na usmjerivačkim sučeljima najbližim izvoru prometa, što nam pak pomaže u tome da se odmah, na samom izvoru, riješimo neželjenih paketa, bez da oni prolaze kroz cijelu mrežu kako bi tek na kraju bili odbačeni. Na ovaj nači štedimo slobodnu širinu pojasa i procesorske kapacitete na usmjerivačima.

Cisco IOS CLI naredbe za *Extended* tj. proširene ACL

Format naredba za kreiranje proširene access liste:

access-list <100-199><deny | permit><protocol><source ip address><wildcard bits><destination ip address><wildcard bits><operator><port or service>

access-list <100-199><deny | permit><protocol> host <source ip address> host <destination ip address><operator><port or service>

access-list <100-199><deny | permit><protocol><source ip address><wildcard bits><destination ip address><wildcard bits>

Primjeri konfiguracije proširene access liste:

Zabrani ili dozvoli pristup izvorišnoj mreži klase C nekoj odredišnoj mreži klase C:

```
router(config)#access-list 100 deny ip 192.168.1.0 0.0.0.255 192.168.4.0 0.0.0.255
router(config)#access-list 100 permit ip 192.168.2.0 0.0.0.255 192.168.4.0 0.0.0.255
```

Zabrani ili dozvoli nekom uređaju (host) pristup odredišnoj /24 mreži:

```
router(config)#access-list 100 deny ip 192.168.1.100 0.0.0.0 192.168.4.0 0.0.0.255
router(config)#access-list 100 deny ip host 192.168.1.100 192.168.4.0 0.0.0.255
router(config)#access-list 100 permit ip 192.168.1.101 0.0.0.0 192.168.4.0 0.0.0.255
router(config)#access-list 100 permit ip host 192.168.1.101 192.168.4.0 0.0.0.255
```

Zabrani ili dozvoli promet bilo kojeg uređaja koji koristi port 80 (http):

```
router(config)#access-list 100 deny tcp any any eq 80
router(config)#access-list 100 permit tcp any any eq 80
```

Zabrani ili dozvoli promet svih uređaja:

```
router(config)#access-list 100 deny any any
router(config)#access-list 100 permit any any
```

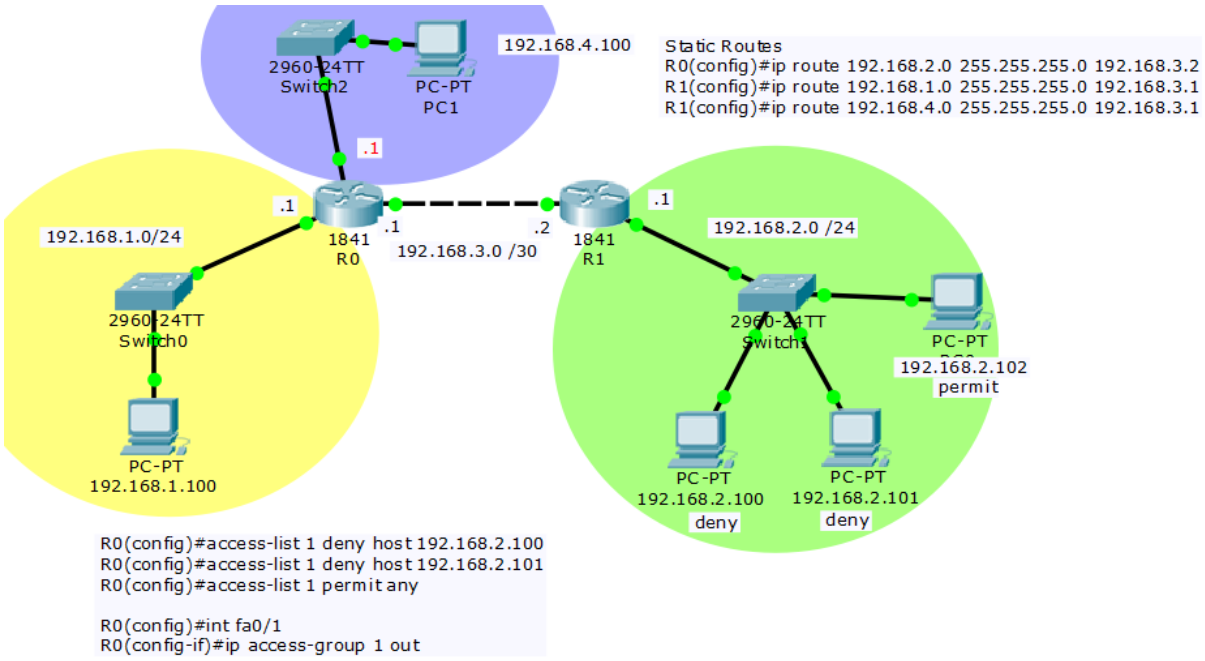
Primjeni pristupnu listu za ulazni i izlazni promet na nekom sučelju

```
router(config)#interface fastethernet 0/0
router(config-if)#ip access-group 100 out

router(config)#interface fastethernet 0/1
router(config-if)#ip access-group 100 in
```

Zadatak - Uporaba proširenih (*extended*) ACL

Na mrežu iz predhodnog zadatka dodajte treću mrežu 192.168.4.0/24 sa jednim računalom.

Postavite dodatnu statičku rutu na R1 kako bi omogućili međusobnu komunikaciju svih mreža



Slika 13.4. *Mrežna topologija za zadatak s proširenim ACL listama*

Sada ćemo ponoviti prošli zadatak tj. izvršiti ćemo zabranu pristupa računalu 2.100 iz zelene mreže u žutu mrežu, ali umjesto standardne koristiti ćemo proširenu ACL listu.

Prvo moramo poništiti postojeću standardu ACL listu.

R0(config)#no access-list 1– briše postojeću

te maknuti postavljanje te liste na sučelje

R0(config)#int fa0/1
R0(config-if)#no ip access-group 1 out

Sada smo dobili čistu situaciju za rad.

Proširena ACL lista koristi se na sučelju najbliže izvoru prometa. U našem slučaju to je sučelje FA0/1 na usmjerivaču R1. Obzirom da kroz to sučelje „ulazi“ promet koji želimo filtrirati, znamo da listu primjenjujemo na ulazni odnosno „in“ promet.

Idemo sada kreirati listu:

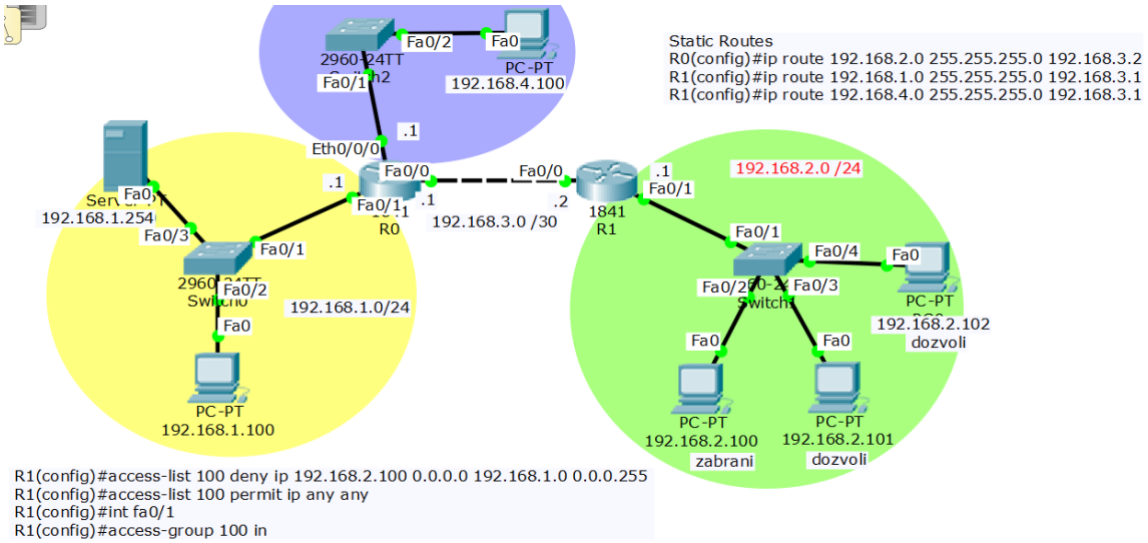
R1(config)#access-list 100 deny ip 192.168.2.100 0.0.0.0 192.168.1.0 0.0.0.255
R1(config)#access-list 100 permit ip any any – svi drugi mogu proći
R1(config)#int fa0/1
R1(config-if)#access-group 100 in - primjena ACL 100 na sučelje i to za ulazni promet

Naredbom „show run“ iz *privilaged* mod provjeri radnu konfiguraciju usmjerivača.

Naredbom „ping“ pokušajte ostvariti vezu sa žutom mrežom. Vidjeti ćete da sva računala osim 2.100 (kojem smo zabranili komunikaciju) imaju vezu.

Idemo sada u žutu mrežu dodati još jedan server koji ima uključen http servis.

Situacija nam je sada kao na slici ispod



Slika 13.5. *Mrežna topologija za zadatak s proširenim ACL listama uz dodani server*

Zadatak: primjenom proširene ACL svima u zelenoj mreži dozvoliti pristup u žutu mrežu ali samo serveru i to samo za http uslugu. Svima iz zelene mreže treba biti omogućena apsolutna komunikacija sa ljubičastom mrežom.

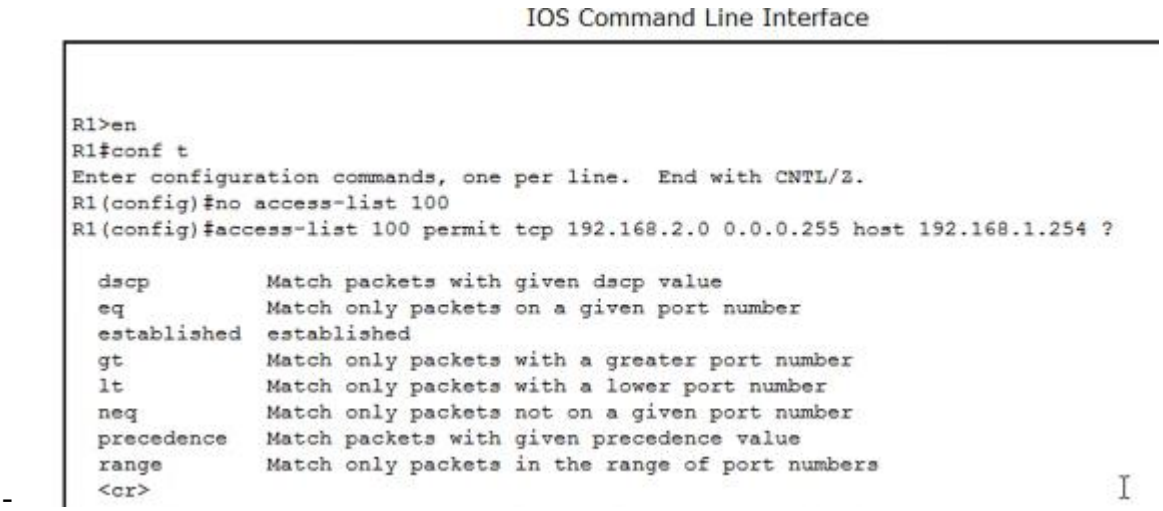
Kako bi obavili ovaj zadatak prvo trebamo pobrisati postojeću ACL 100.

R1(config)#no access-list 100

Sada idemo raditi novu listu. Prva naredba je:

R1(config)#access-list 100 permit tcp 192.168.2.0 0.0.0.255 host 192.168.1.254 eq 80

- Ova naredba nam kaže da se dozvoljava samo tcp protokol iz izvora 192.168.2.0 0.0.0.255 (što je cijela žuta mreža) i to samo prema host 192.168.1.254 (naš server) i to eq 80 (isključivo port 80 koji nam koristi za otvaranje web stranica). Eq je „operand“. Kada u CLI sučelju prilikom upisa naredbe R1(config)#access-list 100 permit tcp 192.168.2.0 0.0.0.255 host 192.168.1.254 ? - dobiti ćete popis mogućnosti. Osim eq (equal tj. jednako) imate i druge opcije npr „veći“, „manji od“,..itd



Slika 13.6. *Opcijske naredbe pri konfiguraciji ACL*

Druga naredba je:

R1(config)#access-list 100 permit ip 192.168.2.0 0.0.0.255 192.168.4.0 0.0.0.255

- Ova naredba dozvoljava potpunu ip komunikaciju između zelene i ljubičaste mreže. Bila je neophodna jer, kao što smo rekli, na kraju svake ACL postoji linija koja se ne vidi ali se podrazumjeva, koja onemogućuje sav promet.

Obzirom da iz predhodnog zadatka već imamo ACL 100 primjenjenu na sučelje, to je sve.

Provjerite rad na način da s nekog računala iz žute mreže u njegov web preglednik postavite direkno adresu servera – trebala bi se otvoriti web stranica. Međutim, ako pokušate kontaktirati web server ili bilo koje drugo računalo npr. naredbom „ping“, to neće biti moguće.

Access liste sa nazivom

ACL lista sa nazivom može biti standardna ili proširena ACL. Jedina razlika je u tome što ih umjesto po broju, prepoznajemo po imenu. IOS naredbe za pokretanje lista sa nazivom su ponešto drugačije od onih koje identificiramo brojem i to na način da naredba započinje sa

„**ip** access-list“ umjesto samo sa „access-list“.

Pogledajmo:

router>enable

router#configure terminal

router(config)#ip access-list <standard | extended><name>

router(config-std-nacl)#<permit | deny><source host or network><wildcard><destination host or network><wildcard> - za standardne ACL sa nazivom

router(config-ext-nacl)#<permit | deny><protocol><source host or network><wildcard><destination host or network><wildcard><operator><port> - za proširene ACL sa nazivom

Zadatak - Primjenom ACL sa nazivom ostvarite istu funkcionalnost kao u prethodnom zadatku.

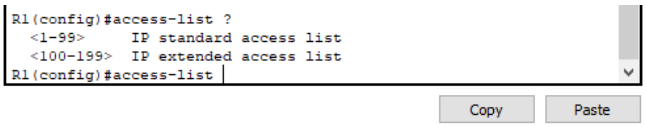
Prva stvar koju trebamo napraviti je uklanjanje postojeće ACL 100 sa usmjerivača, kao i naredbe za primjenu te liste na sučelje usmjerivača.

R1(config)#no access-list 100- briše ACL 100

R1(config)#int fa0/1
R1(config-if)#no access-group 100 in - uklanja primjenu ACL 100 na sučelje

Sada možemo krenuti sa novom konfiguracijom. Ako u CLI naredbu započnemo samo sa

R1(config)#access-list ?

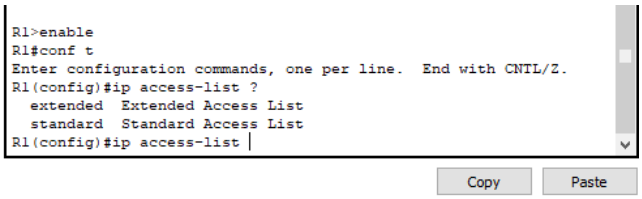


Slika 13.7. *Opcije naredbi pri kreiranju ACL lista*

Vidimo da imamo opciju samo upisa broja – znači ili će biti standardna ili proširena ACL

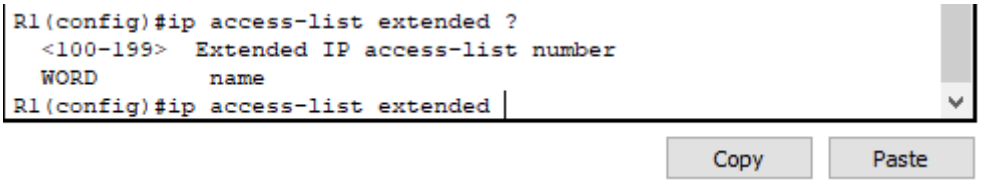
Međutim ako kreiranje ACL započnemo sa

R1(config)#ip access-list ? tada dobijemo opcije kao na slici ispod



Slika 13.8. *Opcije naredbi pri kreiranju ACL lista*

Obzirom da želimo istu funkcionalnost kao i u prošlom zadatku, izabrati ćemo *extended* i opet se poslužiti „?“ da vidimo koje su na sad opcije.



Slika 13.9. *Opcije naredbi pri kreiranju proširenih ACL lista*

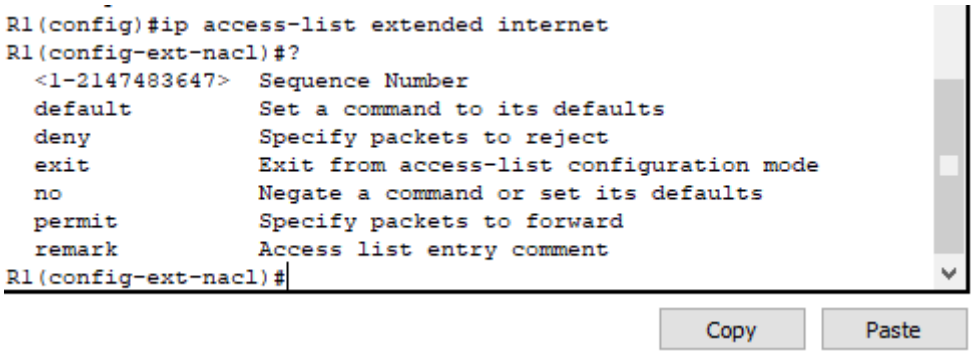
Kao što vidimo, listu možemo nazvati nekim brojem (ali samo onima rezerviranima za proširene ACL) ili pak imenom. Mi ćemo se odlučiti za ime „INTERNET“, pa će nam sada kompletna naredba izgledati

R1(config)#ip access-list extended INTERNET

R1(config-ext-nacl)#

Vidimo da smo sada ušli u konfiguraciju same liste – nešto drugačije nego prije. Idemo pogledati koje su nam sad konfiguracijske opcije

R1(config-ext-nacl)#? i dobijemo prikaz opcija kao na slici ispod



Slika 13.10. *Opcije naredbi pri kreiranju proširenih ACL lista*

Konfiguracija će sada izgledati:

R1(config-ext-nacl)#permit tcp 192.169.2.0 0.0.0.255 host 192.168.1.254 eq 80

R1(config-ext-nacl)#permit ip 192.169.2.0 0.0.0.255 192.168.4.0 0.0.0.255

R1(config-ext-nacl)#exit - završili smo sa konfiguracijom ACL nazvane „INTERNET“

Sada je još treba primjeniti na sučelje:

R1(config)#int fa0/1
R1(config-if)#ip access-group INTERNET in

Sada nam još samo preostaje provjeriti funkcionalnost.

14.Sinteza znanja

Zadatak - Konfiguracija korporativne mreže sa VLAN-ovima, vanjskim i unutarnjim serverima, DHCP-om, DNS-om i Ipsec VPN-om za spajanje djelatnika sa terena na lokalnu mrežu

Slika topologije mreže ovog zadatka zbog svoje veličine nije mogla biti prikazana u ovoj skripti već se može naći na Moodle stranicama predmeta.

Potrebno izvršiti podmrežavanje bloka privatnih IP adresa za dodjelu određenim VLAN-ovima

Unutar korporacijske mreže potrebno je napraviti 4 VLAN-a s mogućnošću međusobnog komuniciranja preko usmjerivača

Da bi VLAN-ovi mogli komunicirati preko usmjerivača neophodno je izvršiti i podmrežavanje (*subneting*). Naime, mi bi mogli napraviti podjelu VLANova i bez podmrežavanja, ali onda im više ne bi mogli omogućiti međusobnu komunikaciju preko usmjerivača.

Znači SVAKI VLAN mora biti u svojoj zasebnoj mreži!

Idemo sada to napraviti.

Prvo preimenujemo preklopnike u SW-Prodaja, SW-Razvoj i SW-Uprava

Sva sučelja na preklopniku su „*default*“ u *Access* modu, pa moramo samo prebaciti mod onih koje želimo postaviti u *trunk* mod.

Možemo to raditi jedno po jedno sučelje kako smo prije naučili, a možemo i proces ubrzati primjenom riječi „range“ unutar naredbi

```
SW1(config)#hostname SW-Prodaja
SW-Prodaja(config)#interface range fastEthernet 0/2 - 3
SW-Prodaja(config-if-range)#switchport mode trunk
SW-Prodaja(config-if-range)#switchport trunk allowed vlan all
```

Nakon što smo kreirali sve *trunk* veze na svim preklopnicima, krećemo dalje i kreiramo sve VLAN-ove.

Dati ćemo primjer samo zadnjeg preklopnika:

```
SW-Uprava>enable
SW-Uprava#conf t
SW-Uprava(config)#vlan 10
SW-Uprava(config-vlan)#name Prodaja
SW-Uprava(config-vlan)#exit
SW-Uprava(config)#vlan 20
SW-Uprava(config-vlan)#name Razvoj
SW-Uprava(config-vlan)#exit
SW-Uprava(config)#vlan 30
SW-Uprava(config-vlan)#name Uprava
SW-Uprava(config-vlan)#exit
SW-Uprava(config)#vlan 40
SW-Uprava(config-vlan)#name Serveri
SW-Uprava(config-vlan)#exit
```

```
SW-Uprava(config)#interface range fastEthernet 0/5 - 19
SW-Uprava(config-if-range)#switchport mode access
SW-Uprava(config-if-range)#switchport access vlan 30
SW-Uprava(config-if-range)#exit
SW-Uprava(config)#vlan 40
SW-Uprava(config-vlan)#name SERVERI
SW-Uprava(config-vlan)#exit
SW-Uprava(config)#interface range fastEthernet 0/20 - 24
SW-Uprava(config-if-range)#switchport mode access
SW-Uprava(config-if-range)#switchport access vlan 40
SW-Uprava(config-if-range)#exit
SW-Uprava(config)#exit
SW-Uprava#copy running-config startup-config
```

NA SVIM PREKLOPNICIMA POTREBNO JE KONFIGURIRATI **SVE POSTOJEĆE VLAN-OVE** IAKO MOŽDA NEMA NITI JEDAN PC KOJI ĆE BITI PRIKLJUČEN NA TAJ VLAN.

BEZ TOGA PREKLOPNIK NE BI ZNAO SVE TRUNKING OZNAKE!

Ne zaboravite snimiti konfiguracije kada završite sa kreiranjem VLAN-ova.

Sada moramo izvršiti konfiguraciju na Router0 da bi VLAN-ovi mogli međusobno komunicirati

```
Router>enable
Router#conf t
Router(config)#interface fastEthernet 0/0
Router(config-if)#no ip address      -da budemo sigurni ako je slučajno bila upisana
Router(config-if)#no shutdown
Router(config-if)#exit
```

Sada moramo napraviti 4 logička sučelja na ovom jednom fizičkom – za svaki VLAN po jedno

```
Router(config)#interface fastEthernet 0/0.10
Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip address 172.16.10.1 255.255.255.0
Router(config-subif)#exit
Router(config)#
Router(config)#interface fastEthernet 0/0.20
Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip address 172.16.11.1 255.255.255.192
Router(config-subif)#exit
Router(config)#interface fastEthernet 0/0.30
Router(config-subif)#encapsulation dot1Q 30
Router(config-subif)#ip address 172.16.11.65 255.255.255.224
Router(config-subif)#exit
Router(config)#interface fastEthernet 0/0.40
Router(config-subif)#encapsulation dot1Q 40
Router(config-subif)#ip address 172.16.11.97 255.255.255.240
Router(config-subif)#exit
```

Kofiguraciju dodjeljenih dhcp adresa na usmjerivaču možemo provjeriti iz *privileged* moda naredbom „show ip dhcp binding“ – i dobiti ćemo prikaz svih trenutno dodjeljenih IP adresa.

Sada konfigurirajte preostale veze na usmjerivačima Router0 i Router1 (ovaj dio već trebate dobro znati te ga nećemo opisivati)

Router1 preimenujte u „ISP“

Sada definirajte IP adresu poslužitelja na korporativnom intranetu. I to STATIČKU! ....jer ovdje je riječ o poslužitelju čija IP adresa mora biti nepromjenjiva!

Definirajte mu odmah i adresu DNS poslužitelja (iako ga još nismo kreirali) i to adresu poslužitelja na internetu 10.10.10.2

Svim računalima u VLAN-ovima IP adresu postavite na DHCP tj. automatsku dodjelu!

Sada idemo kreirati automatsku dodjelu IP adresa u korporativnom intranetu. Tu postoji nekoliko stotina PC-eva i bilo bi teško svakome dodjeljivati statičku IP adresu.

Automatsku dodjelu (DHCP) kreirati ćemo na Router0. Potrebno je napraviti 3 grupe (*pool-a*) adresa, za svaku podmrežu po jednu grupu.

Poslužiteljima moramo statički dodijeliti adrese jer one se ne smiju mjenjati!!!

```
Router(config)#ip dhcp pool VLAN10
Router(dhcp-config)#network 172.16.10.0 255.255.255.0
Router(dhcp-config)#default-router 172.16.10.1
Router(dhcp-config)#dns-server 10.10.10.2
Router(dhcp-config)#exit
Router(config)#
Router(config)#ip dhcp pool VLAN20
Router(dhcp-config)#network 172.16.11.0 255.255.255.192
Router(dhcp-config)#default-router 172.16.11.1
Router(dhcp-config)#dns-server 10.10.10.2
Router(dhcp-config)#exit
Router(config)#ip dhcp pool VLAN30
Router(dhcp-config)#network 172.16.11.64 255.255.255.224
Router(dhcp-config)#default-router 172.16.11.65
Router(dhcp-config)#dns-server 10.10.10.2
Router(dhcp-config)#exit
```

Snimite konfiguraciju usmjerivača!

```
Router#copy running-config startup-config
```

Sada provjerite IP konfiguracije računala!

Sva računala trebala bi dobiti ispravne IP adrese i trebala bi moći komunicirati!

Idemo sada obraditi spajanje na Internet, *wireless connectivity* i DNS!

Kofigurirati ćemo *wireless access point* kojim ćemo spajati laptop na mrežu, kreirati ćemo *default* statičku rutu (0.0.0.0) kojom ćemo promet iz korporativne mreže usmjeravati na Internet, kreirati ćemo statičku rutu za ulaz Internet prometa u našu mrežu, te ćemo kreirati web i DNS poslužitelje.

Prvo ćemo se pozabaviti bežičnom pristupnom točkom (*wireless access point*).

Kada ga označimo u CPT-u, vidimo da ima 2 sučelja (*port*), i to žični i bežični. Žični ne moramo konfigurirati, on je već konfiguriran i samo treba provjeriti da li je na pravoj brzini prijenosa. Ono što moramo konfigurirati je bežićni (*wireless*) port.

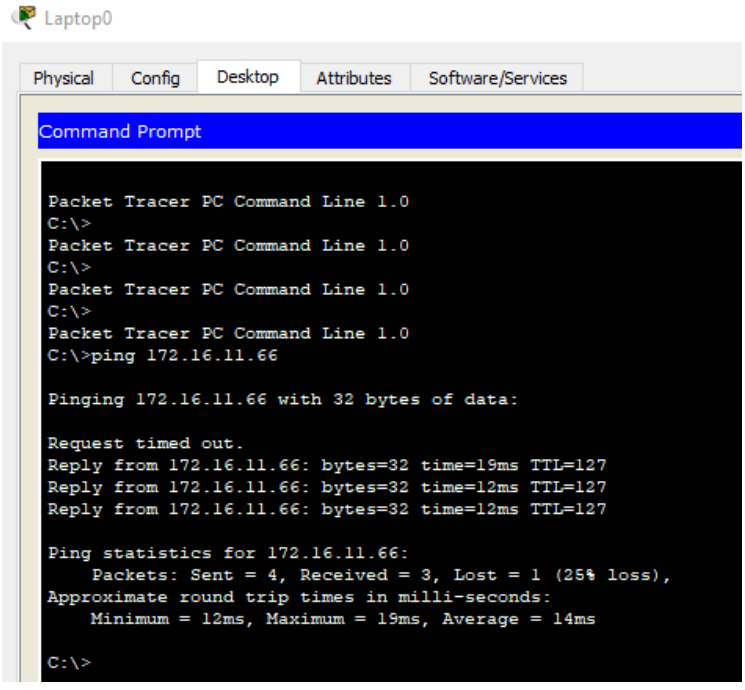
Prvo je potrebno konfigurirati SSID. To u biti predstavlja bežićno IME mreže. Mi ćemo mu dati ime PRODAJA jer je vezan za taj LAN. Sada ćemo mu odrediti način autentifikacije WPA2-PSK (*Pre Shared Key*) sa ključem „sirokopojasne“, sa načinom enkripcije AES (*Advanced Encription Standard*).

Autentifikacija – provjera da je to baš ono računalo koje želimo spojiti. U ovom slučaju koristimo PSK (*Pre Shared Key*) što znači da smo prethodno odredili lozinku i stavili je u konfiguraciju AP-a i prijenosnika. Lozinka je „sirokopojasne“ (u stvarnom životu treba koristiti fraze koje nisu u riječniku, najmanje 8 znakova uz uporabu brojeva i ostalih znakova kako bi lozinku bilo teže za otkriti)

Enkripcija – kodiranje podataka u prijenosu (ukoliko ih neko presretne, ne može ih pročitati)

Sada idemo na prijenosnik (*laptop*). Prvo mu treba zamjeniti mrežnu karticu i staviti PT-LAPTOP-NM-1W module koji ima *wireless* na sebi. Sada mu treba postaviti SSID naše mreže, postaviti ga na WPA2-PSK i dati mu PSK tj. lozinku „sirokopojasne“. Nakon ovoga, prijenosnik bi automatski trebao dobiti IP adresu i trebali bi vidjeti na ekranu uspostavljenu *wireless* konekciju. Ukoliko se to odmah ne dogodi, prebacite dodjelu IP adrese na statički način, pa ga opet vratite na automatski (DHCP)...ponovite nekoliko puta dok se ne pojavi IP adresa. Sada ponovo provjeri na usmjerivaču dodjeljene DHCP IP adrese naredbom „show ip dhcp binding“ i trebali bi vidjeti 4 adrese.

Probajte sada sa laptopa „pingati“ neko računalu u drugom LAN-u. („ping“ iz *CMD prompt* sučelja da vidite sve odzive).



Slika 14.1. *Rezultati izvršenja „ping“ naredbe*

Vidjet ćete da prvi „ping“ nije bio uspješan, tj. da je *timed out*. Zašto se to događa?

Računalo (u oveme slučaju naš prijenosnik) zna da adresa na koju šalje „ping“ nije u njegovoj podmreži (to zna pomoću svoje vlastite IP adrese i *subnet* maske), te da stoga „ping“ paket mora poslati prema svom *default gateway*-u. Paket će sada doći do usmjerivača, koji će pogledati gdje je odredište paketa, shvatiti će da je tu u LAN-u koji je također preko *trunk* veze spojen DIREKTNO na njega, te će proslijediti paket u taj subnet tj. u taj LAN do odredišnog uređaja. Kako bi to bilo moguću, Ethernet okvir mora imati postavljenu odredišnu MAC adresu. Međutim, prijenosnik u tom prvom trenutku nezna MAC adresu računala prema kojem treba uputiti paket jer još nikada prije nisu međusobno komunicirali. Obzirom da prvo treba saznati MAC adresu, računalo prvo šalje tzv. ARP poruku, kako bi saznalo MAC adresu.....iz tih razloga dozvoljeno vrijeme za samu „ping“ naredbu istekne. Slijedeće ping naredbe prolaze jer je sad već poznata MAC adresa odredišta.

Probajmo sada „pingati“ server koji se nalazi na internetu (10.10.10.2) i vidjeti ćemo da veza ne prolazi. Zašto? Zato što naš *gateway* usmjerivač nezna kamo preusmjeriti paket za tu adresu.

U situaciji kao na slici mi bismo mogli pokrenuti neki od dinamičkih protokola usmjeravanja na našem *gateway* usmjerivaču, ali to ne želimo raditi jer bi inače svima obznanili strukturu naše unutarnje mreže. (a to i ne smijemo jer unutra koristimo privatne adrese). Iz tih razloga, na našem gateway usmjerivaču pokrenuti ćemo **statičku *default* rutu** za sav odlazni promet koji ne završava u našoj korporativnoj mreži.

To radimo jednostavnom naredbom

```
Router>enable
Router#conf t
Router(config)#ip route 0.0.0.0 0.0.0.0 serial 0/0/0
Router(config)#
```

Na ovaj način riješili smo odlazni pravac. Sada treba omogućiti dolazni pravac. To ćemo napraviti postavljanjem **statičke rute** na ISP usmjerivaču. Obzirom da smo sve podmreže na našoj korporativnoj mreži dobili iz jednog velikog seta adresa koje smo subnetirali, ta velika mreža u biti sadrži sve naše podmreže. Ukoliko adresu velike mreže postavimo kao odredišnu mrežu, paketi koji trebaju ići u bilo koju od naših podmreža biti će proslijeđeni tom adresom sve do našeg usmjerivača koji razlikuje podmreže (Router0), i tek će on napraviti razvrstavanje na podmreže i poslati paket samo u onaj LAN u koji treba.

Naredba kojom postavljamo statičku rutu na ISP usmjerivač je

```
ISP>enable
ISP#conf t
ISP(config)#ip route 172.16.10.0 255.255.254.0 88.40.12.1
```

Kao što vidite, ovaj put smo koristili adresu odredišnog sučelja iako smo mogli umjesti 88.40.12.1 koristiti i serial 0/0/0

Sada probajte ponovno „pingati“ web server na Internetu i vidjeti ćete da veza prolazi. Opet će prvi „ping“ biti *timed out* iz objašnjenih razloga, a ostali će biti uspješni.

```
Pinging 10.10.10.2 with 32 bytes of data:

Request timed out.
Reply from 10.10.10.2: bytes=32 time=12ms TTL=126
Reply from 10.10.10.2: bytes=32 time=18ms TTL=126
Reply from 10.10.10.2: bytes=32 time=37ms TTL=126

Ping statistics for 10.10.10.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 12ms, Maximum = 37ms, Average = 22ms

C:\>
```

Slika 14.2. *Rezultati izvršenja „ping“ naredbe*

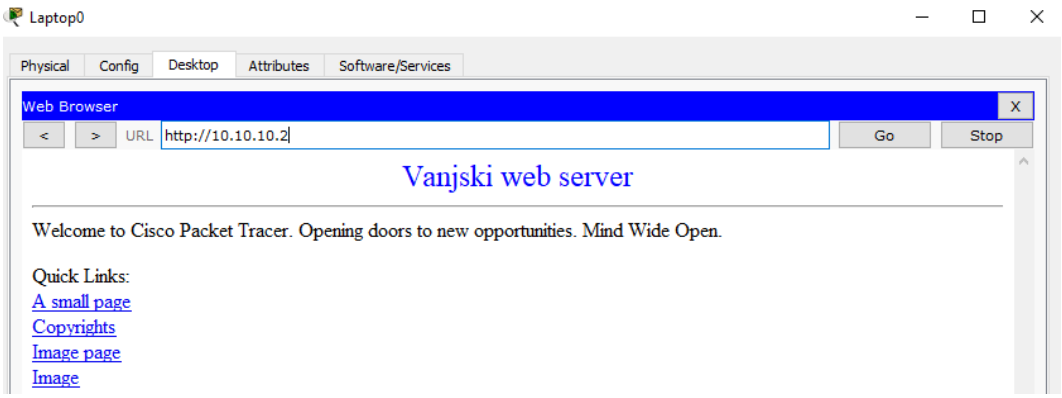
Sada ponovno snimite konfiguracije oba usmjerivača.

Idemo sada pokrenuti web poslužitelj na Internetu.

Čim uđemo na njega, idemo na SERVICES tab pa HTML i vidjeti ćemo da je *web server* funkcija već pokrenuta, te da se na poslužitelju nalazi nekoliko datoteka. Odaberemo onu sa imenom „index.html“ i promjenimo malo teksta kako bismo bili sigurni da je to baš server kojeg želimo vidjeti....npr. umjesto „Cisco Packet Tracer“ napišemo „Vanjski web server“.

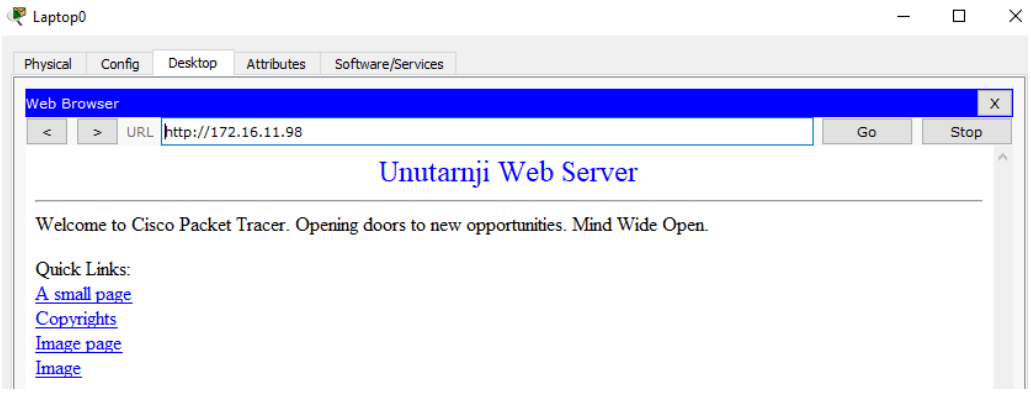
Sada sa našeg prijenosnika pokrenemo web preglednik (*browser*) i kao odredišnu adresu postavimo 10.10.10.2

Vidimo da se otvorila naša tražena web stranica!



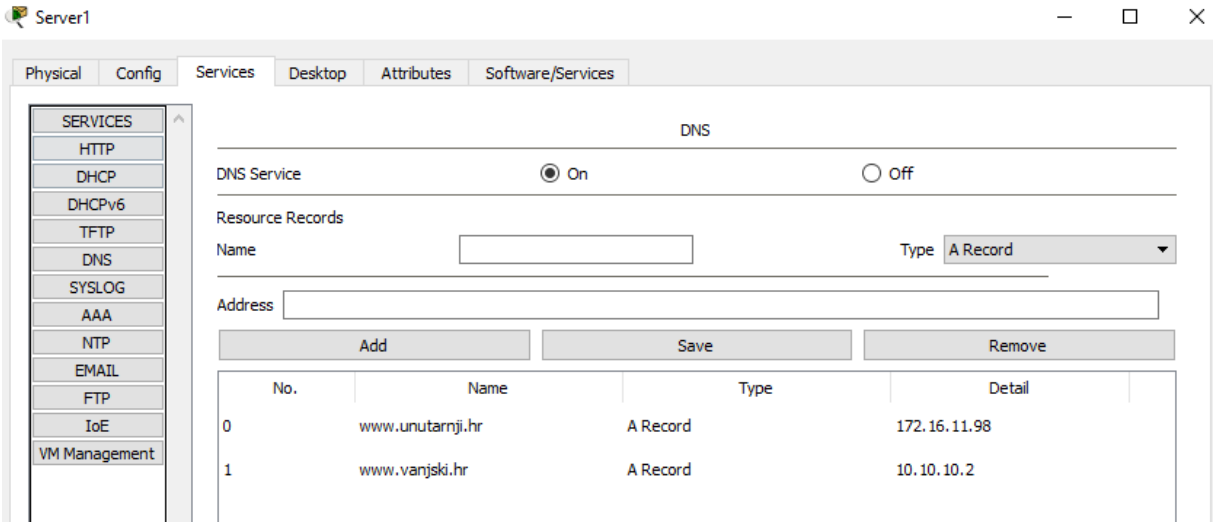
Slika 14.3. *Dohvat vanjskog web servera*

Sada na isti način pokrenemo unutarnji web poslužitelj koji se nalazi na našoj korporativneoj mreži, te u natpisu postavimo tekst „Unutarnji Web Server“. Provjerimo rad tako da sa prijenosnika kroz *Web Browser* otvaramo stranicu sa adrese http://172.16.11.98 i vidjeti ćemo da je i ova stranica dostupna.



Slika 14.4. *Dohvat unutarnjeg web servera*

Sada nam je još preostalo pokrenuti DNS server kako bi mogli posjećivati web stranice bez da znamo njihove IP adrese već nama jednostavnija „ljudska imena web stranica“. Obzirom da smo već prije kroz DHCP svima obznanili IP adresu DNS servera (10.10.10.2), sada samo moramo na tom serveru pokrenuti DNS uslugu, te kreirati DNS zapise za naše dvije web stranice.



Slika 14.5. *Aktivacija DNS usluge*

Provjerimo sada rad, na način da sa prijenosnika pokušamo otvoriti web stranice, ali ne pomoću postavljanja IP adresa u Web brawser već postavljanjem imena stranica www.unutarnji.hr i www.vanjski.hr

Vidimo da i to radi!

Naša mreža je trenutno potpuno nezaštićena. Svak se može priključiti na bilo koji preklopnik ili usmjerivač, te mjenjati konfiguracije po volji. Također, sa svakog računala može se pristupiti svakom drugom računalu.

Slijedeći korak je postavljanje sigurnosnih postavki!

Ovdje ćemo se baviti:

- postavljanjem korisnika njihovih lozinki, te razine pristupa

- konfiguracijom enkripcije lozinke, blokade prijave na mrežu te postavljanjem natpisa
- konfiguracijom daljinskog pristupa mreži putem Telnet (loše) i SSH (puno bolje) pristupa
- konfiguracijom *Switchport Security*-a (tj zaštite da se na sučelje ne spaja računalo kojem to nije dozvoljeno)

Ukoliko se priključimo na Router0 u *user* modu malo toga možemo napraviti, međutim već naredbom „enable“ i ulaskom u *privileged* mod dobijamo veće ovlasti.

Naredbom „show privilege“ dobijamo podatak da smo na razini 15 što je maksimalna razina. Što se može raditi u određenom razredu privilegija već je tvornički zadano. U sklopu ovog kolegija nećemo ulaziti u detalje zaštite.

Pristup usmjerivaču u CPT kroz CLI u biti simulira pristup preko konzolnog sučelja (većina usmjerivača ima jedan konzolni ulaz), pa ćemo za početak postaviti lozinku za pristup preko konzolnog sučelja.

```
Router>enable
Router#conf t
Router(config)#line console 0
Router(config-line)#password hajdukjeprvak2018
Router(config-line)#login
Router(config-line)#exit
```

Ukoliko želimo maknuti lozinku ili je promjeniti idemo

```
Router>enable
Router#conf t
Router(config)#line console 0
Router(config-line)#no password hajdukjeprvak2018
```

Sada ćemo obraditi temu daljinskog pristupa mreži (*remote access VPN*)

Prvo ćemo u CPT dodati udaljeno računalo, koje je na Internet spojeno DSL modemom i Cloud koji će predstavljati Internet vezu između našeg ulaznog usmjerivača i tog DSL modema.

Računalo spjamo na „port 1“ na modemu, a „port 0“ na modemu (ima oznaku telefonske linije) spajamo na Cloud jer predstavlja telefonsku paricu kojom DSL tehnologija spaja korisnike i mrežu ISP-a. Oznaka paričnog kabela (*Phone*) koji koristimo za taj spoj je „isprekidana crna munja“.

Vezu Clouda i našeg ulaznog usmjerivača predstavljati će FE veza.

Na DSL modemu ne moramo ništa konfigurirati, ali na Cloudu moramo. Idemo na DSL TAB i moramo dodati vezu „Modem4 Ethernet6“ u konfiguraciju (samo pritisni ADD)

Sada trebamo dodjeliti neke globalne adrese. Uzeti ćemo blok adresa 72.44.20.0/28 (ukupno 16 adresa od kojih možemo koristiti 14. Prvu ćemo uzeti za usmjerivač 72.44.20.1 , a zadnja za uređaje je .14)

Dodjelu i tih adresa prepustiti ćemo DHCP-u.

Prvo dodjeljujemo adresu sučelju usmjerivača

```
Router>enable
Router#conf t
Router(config)#interface fastEthernet 0/1
Router(config-if)#ip address 72.44.20.1 255.255.255.240
Router(config-if)#no shutdown
```

Sada pokrećemo DHCP za vanjske adrese

```
Router(config)#ip dhcp pool remote_pool
Router(dhcp-config)#network 72.44.20.0 255.255.255.240
Router(dhcp-config)#default-router 72.44.20.1
Router(dhcp-config)#dns-server 10.10.10.2
```

Sada na PC-u postavi dodjelu IP adrese na DHCP i trebali bi dobiti adresu! Snimite konfiguraciju usmjerivača!

Sa vanjskog PC-a trebali bi moći „pingati“ sve u korporativnoj mreži, ali ne i Vanjski web server i to zbog povratne rute između našeg usmjerivača i ISP usmjerivača koji vraća „ping“ samo u 172.16.10.0/23 mrežu, ali ne i u ovu novu vanjsku!

Sada idemo konfigurirati VPN

Prvo moramo konfigurirati AAA (*Authorisation Authentification Accounting*) uslugu kako bi mogli odrediti tko se smije priključiti na naš korporativni VPN.

```
Router#enable
Router#conf t
Router(config)#aaa new-model    -kreira AAA set pravila zvan „new-model“
Router(config)#aaa authentication login REMOTE local    - postavlja autentifikaciju za prijavu pomoću liste „REMOTE“ koju imamo spremljenu lokalno na ovom usmjerivaču
Router(config)#aaa authorization network REMOTE local  – postavlja autorizaciju preko iste liste
Router(config)#username VPN secret supersecure   -lokalno kreiranje korisnika
```

Sada trebamo kreirati ISEKMP kriptiranje

```
Router(config)#crypto isakmp policy 10
Router(config-isakmp)#encryption aes 256
Router(config-isakmp)#hash md5
Router(config-isakmp)#authentication pre-share
Router(config-isakmp)#group 2
Router(config-isakmp)#lifetime 21600
Router(config-isakmp)#exit
```

ISAKMP PHASE 1

Sada idemo kreirati grupu korisnika

```
Router(config)#crypto isakmp client configuration group REMOTE
Router(config-isakmp-group)#key cisco
Router(config-isakmp-group)#pool MYPOOL
```

Sada kreiramo pool adresa za dodjelu VPN korisnicima. Uzimamo iz VLAN-a Prodaja i to 50 adresa iz viših adresa kako bi bila manja mogućnost interferencije sa računalima u VLAN-u

```
Router(config)#ip local pool MYPOOL 172.16.10.150 172.16.10.200
```

Sada idemo kreirati ISAKMP Phase 2

```
Router(config)#crypto ipsec transform-set MYSET esp-aes 256 esp-md5-hmac
```

Sada kreiramo dinamičku mapu

```
Router(config)#crypto dynamic-map DYNMAP 10
```

```
Router(config-crypto-map)#set transform-set MYSET
Router(config-crypto-map)#reverse-route     - automatski insertira statičku rutu
Router(config-crypto-map)#exit
```

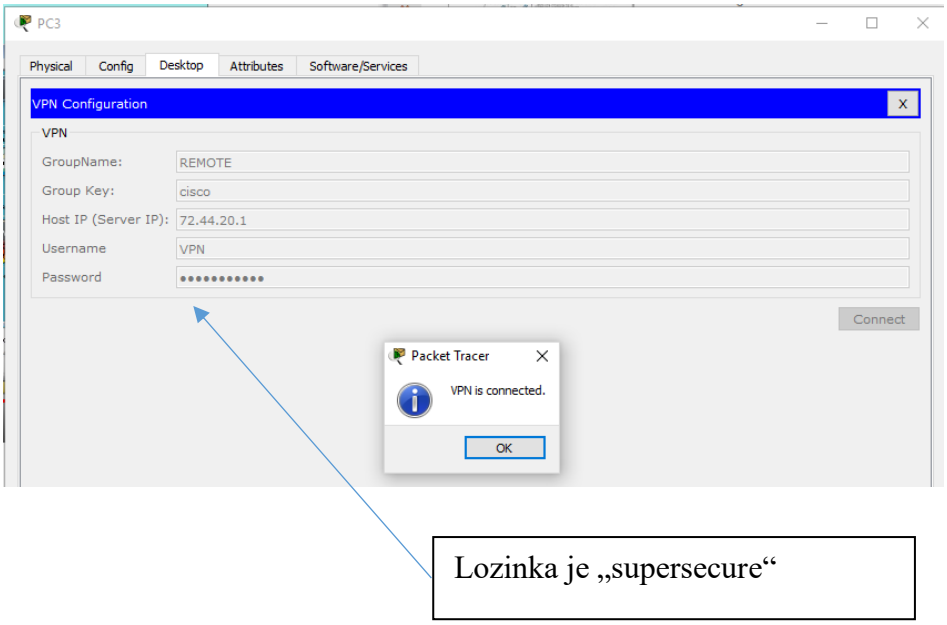
Sada idemo na izradu kripto mape

```
Router(config)#crypto map CLIENT_MAP client authentication list REMOTE
Router(config)#crypto map CLIENT_MAP isakmp authorization list REMOTE
Router(config)#crypto map CLIENT_MAP client configuration address respond
Router(config)#crypto map CLIENT_MAP 10 ipsec-isakmp dynamic DYNMAP
```

Sada idemo primjeniti kripto mapu na sučelje

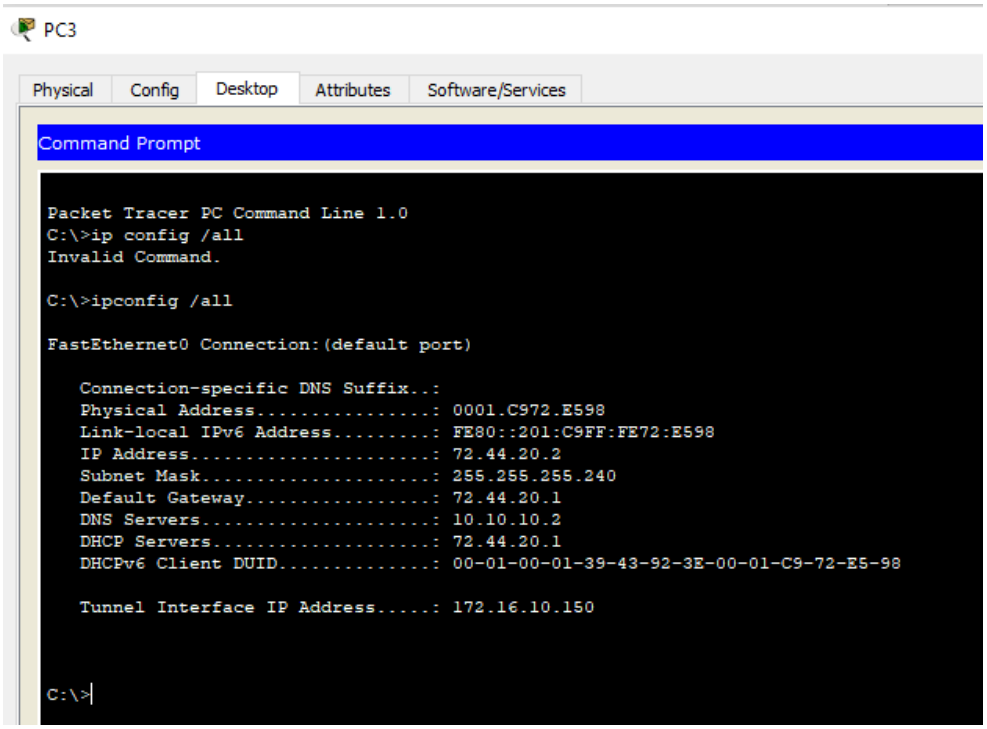
```
Router(config)#interface fastEthernet 0/1
Router(config-if)#crypto map CLIENT_MAP
Router(config-if)#exit
```

Sada trebamo podesiti VPN vezu na udaljenom računalu. Naime, mi trenutno imamo vezu tog PC-a (možemo pingati sva računala u korporativnoj mreži), ali on i dalje ima nezaštićenu vezu



Slika 14.6. *Konfiguracija VPN-a na PC3*

Ako sada idemo na CMD i pogledamo ip konfiguraciju računala, vidjeti ćemo tunelsko sučelje!



Slika 14.7. *Pregled svih sučelja na PC3*

Ako sada idemo na naš usmjerivač i sa naredbom provjerimo „isakmp security assosiation“

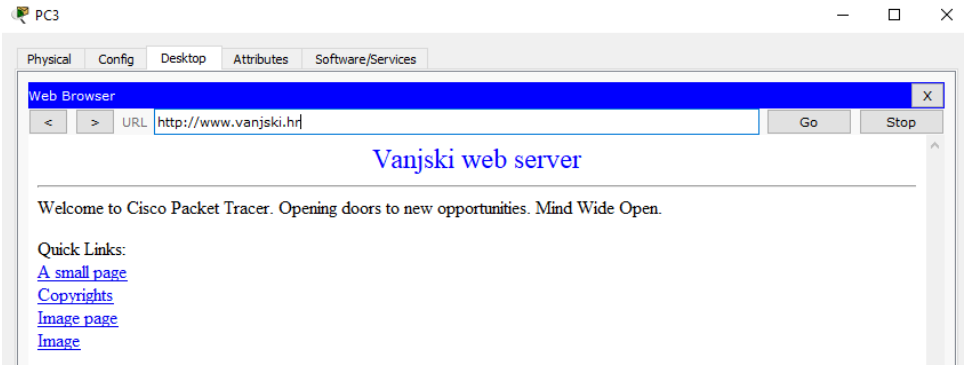
```
Router#show crypto isakmp sa
```

IPv4 Crypto ISAKMP SA

dst	src	state	conn-id slot status
72.44.20.2	72.44.20.1	QM_IDLE	1080 0 ACTIVE

IPv6 Crypto ISAKMP SA

Vidimo da je VPN veza aktivna! Sada naše udaljeno računalo izgleda kao da je priključeno u lokalnu mrežu. Možemo otvarati web stranice sa unutarnjeg i vanjskog servera koristeći nazive stranica (jer nam radi DNS)

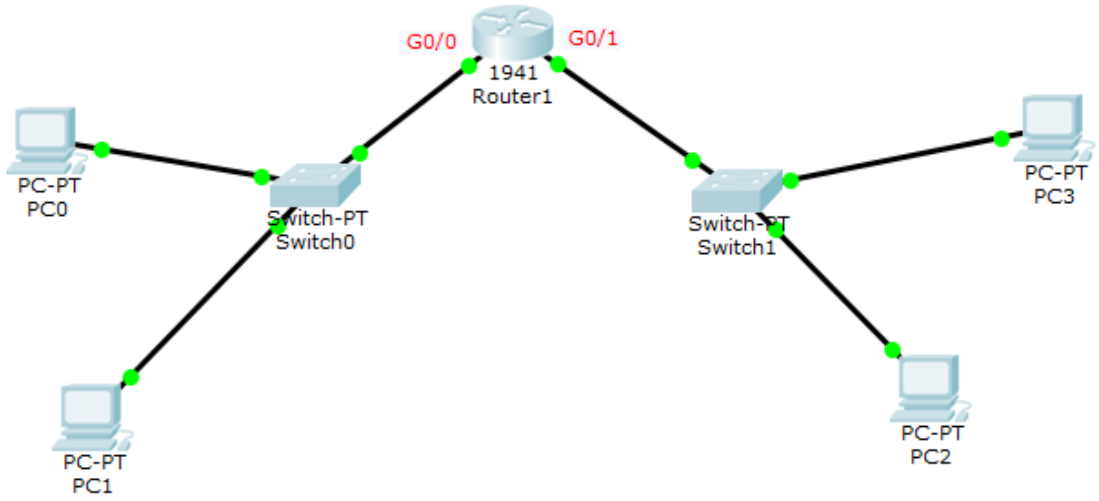


Slika 14.8. *Dohvat vanjskog web servera sa PC3*

15. Adresiranje u IPv6

Zadatak 1 - Upotreba jednoodredišne adrese za lokalno korištenje (*link local*)

Na prikazanoj mreži provjeriti jednoodredišne adrese za lokalno korištenje (*link local*) računala, konfigurirati usmjerivač za podršku Ipv6 te konfigurirati sučelja za lokalno korištenje. Provjeriti funkcionalnost naredbom „ping“.



Slika 15.1. *Mrežna topologija za zadatak 1*

Upute

Komunikacija dvaju susjednih čvorova koji se nalaze na istoj poveznici (LAN-u) odvija se pomoću adresa lokalne poveznice (eng. *link local address*). Adresa lokalne poveznice potrebna je za proces otkrivanja susjeda (eng. *Neighbor Discovery*) i na računalima se uvijek postavlja automatski.

- **Adresa lokalne poveznice** uvijek započinje sa sekvencom FE80, uz prefiks **FE80::/64** (identifikator mrežnog sučelja veličine je 64 bita).
- 64 bitni identifikator mrežnog sučelja kreira se automatski pomoću MAC adrese

Promet IPv6 paketa koji sadrže adresu lokalne poveznice nikad se od strane usmjerivača ne prosljeđuje izvan lokalne poveznice

```
C:\> ipconfig /all

FastEthernet0 Connection:(default port)

   Connection-specific DNS Suffix..:
   Physical Address.....................: 00E0.8F11.953E
   Link-local IPv6 Address.....: FE80::2E0:8FFF:FE11:953E
   IP Address.....................: 0.0.0.0
   Subnet Mask.....................: 0.0.0.0
   Default Gateway.....................: 0.0.0.0
   DNS Servers.....................: 0.0.0.0
   DHCP Servers.....................: 0.0.0.0
   DHCPv6 Client DUID.....: 00-01-00-01-BC-E3-55-7D-00-E0-8F-11-95-3E
```

Slika 15.2. *Prikaz svih sučelja računala*

Konfiguracija usmjerivača:

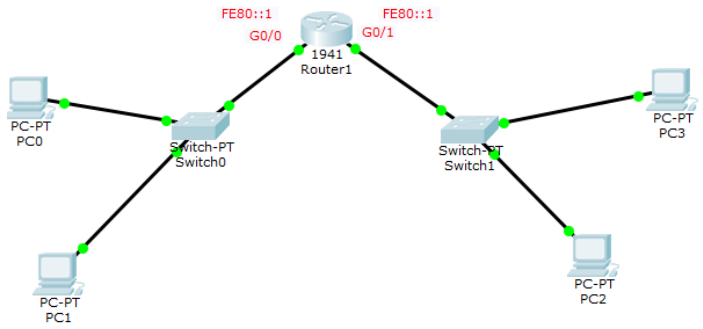
```
Router>enable
Router#configure terminal
Router(config)#ipv6 unicast-routing        -Aktivacija IPv6 usmjeravanja

Router(config)#interface g0/0
Router(config-if)#ipv6 address FE80::1 link-local -Dodjela lokalne Ipv6 adrese G0/0
Router(config-if)#no shutdown

Router(config)#interface g0/1
Router(config-if)#ipv6 address FE80::1 link-local  Ista adresa kao i G0/0 – Dozvoljeno!
Router(config-if)#no shutdown
```

Iste lokalne adrese sučelja u Ipv6 su dozvoljene jer se promet adresiran na taj način NIKADA ne isporučuje van lokalne poveznice.

Uz pomoć naredbe „ping“ i adresiranja uz pomoć *jednoodredišne adrese za lokalno korištenje (link local)* provjeri veze između računala, te računala i sučelja na usmjerivaču.

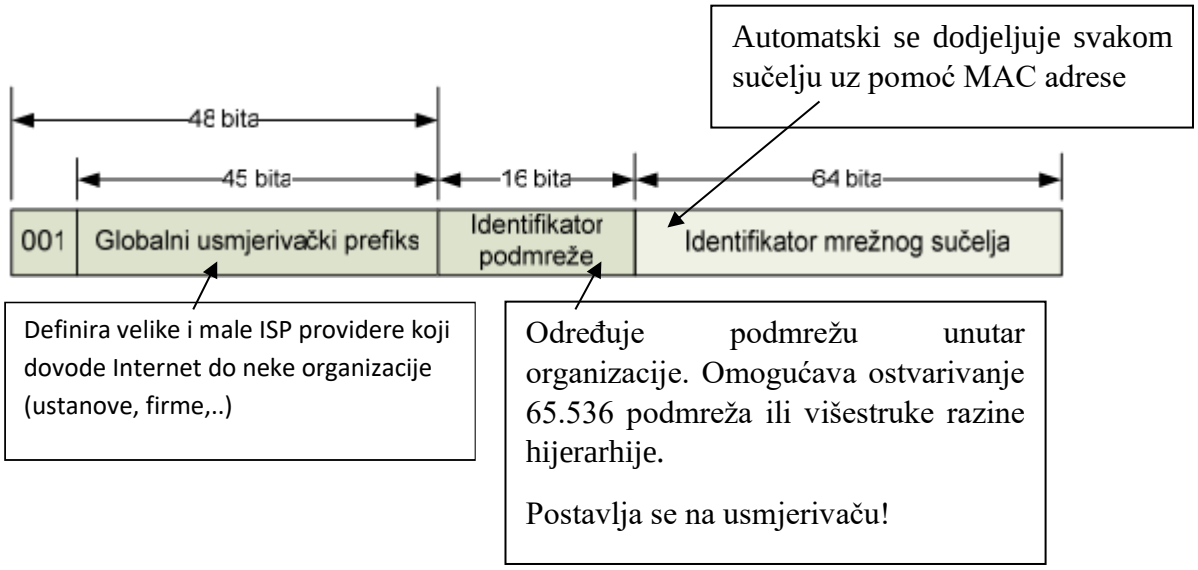
Zadatak 2- Uporaba globalne adrese za jednoodredišno adresiranje

Na postavkama iz prošlog zadatka omogućiti međusobnu komunikaciju svih računala uz pomoć Ipv6 podmrežavanja i dodjele globalnih jednoodredišnih adresa.



Slika 15.3. *Mrežna topologija za zadatak 2*

Upute: Globalne jednoodredišne adrese istovjetne su javnim IPv4 adresama. To su adrese koje su dostupne na globalnoj razini. Doseg adrese jest cjelokupna IPv6 mreža (Internet). Fiksni dio postavljen na vrijednost 001 pa je prefiks za globalne adrese 2000::/3



Prva 64 bita globalne jednoodredišne adrese nazivaju se „mrežni dio“ adrese, a preostala 64 bita su dio za sučelja (host part) [2]

I----mrežni dio----------I-------host dio----------I

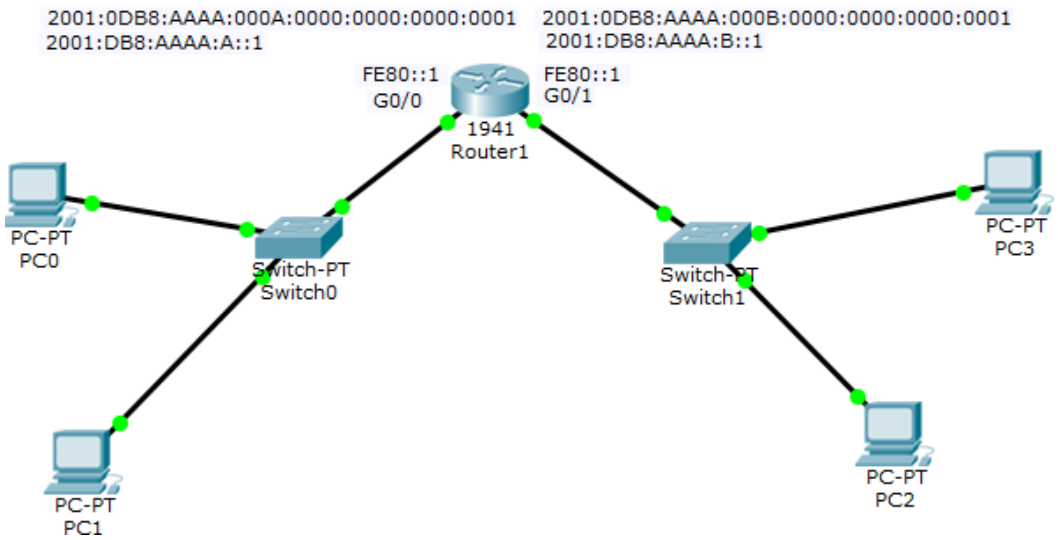
2001:0DB8:AAAA:000A:0000:0000:0000:0000

Dodjeljuje IANA (Internet Assigned Number Authority)

Dodjeljuje ISP za npr. za vezu sa našom organizacijom

Dodjeljuje mrežni administrator unutar naše organizacije za dodjelu pod-mreža

Za svrhu ove vježbe uzeti ćemo proizvoljni mrežni dio globalne Ipv6 adrese 2001:0DB8:AAAA koji bi nam trebao definirati ISP provider, za podmreže (koje unutar neke firme ili organizacije mi sami kreiramo) odabrati ćemo 000A i 000B za naše dvije podmreže, a svakom sučelju usmjerivača koji vodi prema određenoj podmreži dojeliti ćemo vrijednost 1, tj to je prva adresa u podmreži.



Slika 15.4. *Mrežna topologija sa adresama za zadatak 2*

Konfiguracija usmjerivača:

```
Router>enable
Router#configure terminal
Router(config)#interface g0/0
Router(config-if)#ipv6 address 2001:DB8:AAAA:A::1/64
Router(config-if)#no shutdown
Router(config-if)#interface g0/1
Router(config-if)#ipv6 address 2001:DB8:AAAA:B::1/64
Router(config-if)#no shutdown
```

Sada na svakom računalu treba na „IP config“ sučelju odabrati automatsku IPv6 konfiguraciju, te bi je računalo trebalo samo dobiti od usmjerivača. Testirajte vezu svih računala uz pomoć „ping“ naredbe

Zaključak:

Upotrebom IPv6 protokola znatno se pojednostavi postavljanje adresa na računala, tj. to se obavlja automatski obzirom da računalo svoju IPv6 adresu samo postavlja pomoću mrežnog usmjerivača.

Zadatak 3 - Aktivacija RIP protokola nove generacije (RIPng) u IPv6 mrežama

Cilj ove vježbe je konfiguracija IPv6 mreže sa RIPng protokolom. Potrebno je aktivirati RIPng protokol na svim usmjerivačima, te ispravnoi izvršiti adresiranje svih komponenti.

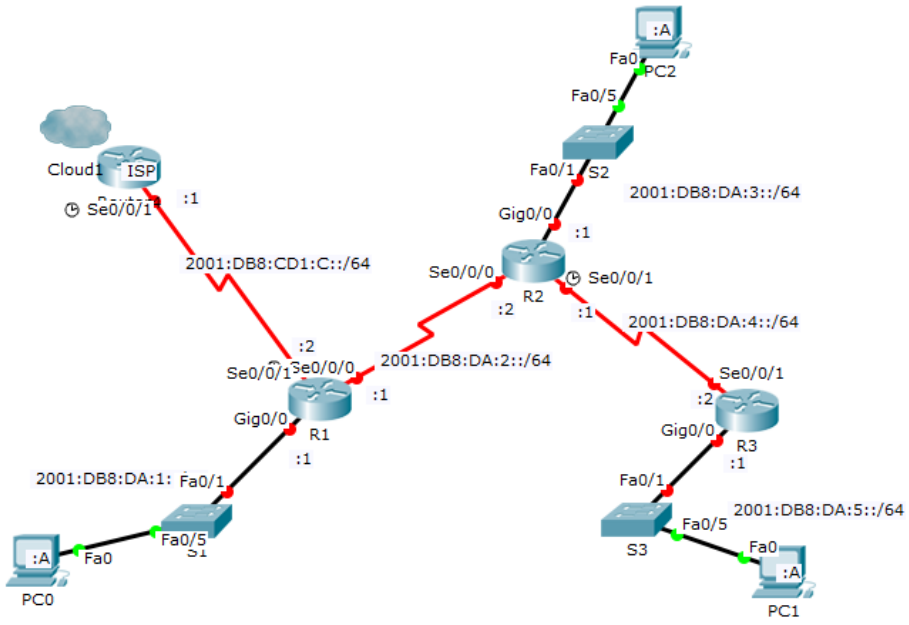
1. Konfigururaj svaki PC sa:
-10-tom adresom iz IPv6 podmreže (npr. :A)
- /64 je *network prefix* duljina
- postavi *default gateway* koristeći usmjerivačku *link-local* adresu,
- *pazi: *link-local* adresa samog računala je već auto-konfigurirana

2. Na svim usmjerivačima postavi slijedeće:
- naziv uređaja (*hostname*) (ISP, R1, R2, R3)
- omogući IPv6 usmjeravanje (*routing*)
- konfiguriraj sva sučelja sa predloženim IPv6 *link-local* adresama,i *global unicast* IPv6 adresama (vidi topologiju mreže)
- *serial DCE interfaces treba clock rate* postavi na 128000

3. Na R1, R2, R3:
- omogući RIPng usmjeravanje na svakom sučelju osim na R1: S0/0/1
- koristi ime RIP1 (velika slova) za ime RIPng procesa

4. Na R1:
- postavi IPv6 *default route* iz s0/0/1 sučelja i oglasi tu rutu ostatku mreže koristeći RIPng

5. Na ISP usmjerivaču:
- postavi IPv6 *summary route*, iz s0/0/1 sučelja,za dohvat svih podmreža (R1, R2, i R3 podmreže)



Slika 15.5. *Mrežna topologija sa adresama za zadatak 3*

Izvedba vježbe:

Prvo svim računalima dodamo *link-lokal* adresu usmjerivača za *default gateway*, te im dodamo njihovu globalnu jednoodredišnu adresu i to statičkom metodom jer još nismo konfigurirali usmjerivače (inače bi mogli samo pokrenuti auto-config)

Konfiguracija usmjerivača R1:

```
Router(config)#hostname R1
R1(config)#ipv6 unicast-routing      -omogućimo IPv6 usmjeravanje
```

Konfiguracija g0/0 sučelja

```
R1(config)#interface g0/0
R1(config-if)#IPv6 address FE80::1 link-local
R1(config-if)#IPv6 address 2001:DB8:DA:1::1/64
R1(config-if)#no shutdown
```

Konfiguracija Se0/0/0 sučelja

```
R1(config)#interface serial 0/0/0
R1(config-if)#IPv6 address FE80::1 link-local
R1(config-if)#IPv6 address 2001:DB8:DA:2::1/64
R1(config-if)#clock rate 128000
R1(config-if)#no shutdown
```

Konfiguracija Se0/0/1 sučelja

```
R1(config)#interface serial 0/0/1
R1(config-if)#IPv6 address FE80::1 link-local
R1(config-if)#IPv6 address 2001:DB8:CD1:C::2/64
R1(config-if)#no shutdown

R1(config-if)#ipv6 route ::/0 s0/0/1      -postavljanje default rute na S0/0/1 sučelju

R1(config-if)#interface g0/0
R1(config-if)#ipv6 rip RIP1 enable  aktivacija RIPng protokola za g0/0
R1(config-if)#interface s0/0/0
R1(config-if)#ipv6 rip RIP1 enable  aktivacija RIPng protokola za s0/0/0
R1(config-if)#ipv6 rip RIP1 default-information originate    propagacija default
rute s0/0/1 pomoću RIPng protokola svim ostalim usmjerivačima
R1#copy running-config startup-config      Spremanje trenutne konfiguracije usmjerivača u
startup konfiguraciju
R1#show run  za provjeru konfiguracije
```

Konfiguracija usmjerivača R2

```
Router(config)#hostname R2
R2(config)#ipv6 unicast-routing
```

Konfiguracija Se0/0/0 sučelja

```
R2(config)#interface serial 0/0/0
R2(config-if)#ipv6 address FE80::2 link-local
R2(config-if)#ipv6 address 2001:DB8:DA:2::2/64
R2(config-if)#no shutdown
```

Konfiguracija g0/0 sučelja

```
R2(config)#interface g0/0
R2(config-if)#ipv6 address FE80::2 link-local
R2(config-if)#ipv6 address 2001:DB8:DA:3::1/64
R2(config-if)#no shutdown
```

Konfiguracija Se0/0/1 sučelja

```
R2(config-if)#interface serial 0/0/1
R2(config-if)#ipv6 address FE80::2 link-local
```

```
R2(config-if)#ipv6 address 2001:DB8:DA:4::1/64
R2(config-if)#clock rate 128000
R2(config-if)#no shutdown

R2(config-if)#ipv6 rip RIP1 enable   Aktivacija RIPng na svim sučeljima
R2(config-if)#interface g0/0
R2(config-if)#ipv6 rip RIP1 enable
R2(config-if)#interface serial 0/0/0
R2(config-if)#ipv6 rip RIP1 enable

R2#copy running-config startup-config        Spremanje trenutne konfiguracije usmjerivača u startup konfiguraciju
```

Ako je sve OK možemo naredbom "show ipv6 route" vidjeti tablicu usmjeravanja

```
R2#show ipv6 route
```

Konfiguracija usmjerivača R3:

Router(config)#hostname R3
R3(config)#ipv6 unicast-routing

Konfiguracija Se0/0/1 sučelja

```
R3(config)#interface serial 0/0/1
R3(config-if)#ipv6 address FE80::3 link-local
R3(config-if)#ipv6 address 2001:DB8:DA:4::2/64
R3(config-if)#no shutdown
```

Konfiguracija g0/0 sučelja

```
R3(config-if)#interface g0/0
R3(config-if)#ipv6 address FE80::3 link-local
R3(config-if)#ipv6 address 2001:DB8:DA:5::1/64
R3(config-if)#no shutdown
R3(config-if)#ipv6 rip RIP1 enable   Aktivacija RIPng na svim sučeljima
R3(config-if)#interface serial 0/0/1
R3(config-if)#ipv6 rip RIP1 enable
R3# copy running-config startup-config Spremanje trenutne konfiguracije usmjerivača u startup konfiguraciju
```

Konfiguracija usmjerivača ISP:

Na njemu ne aktiviramo naš RIPng protokol jer on već spada u vanjsku mrežu već mu informacije o svim mrežama koje su na njega spojene dajemo pomoću „summary route".

```
Router(config)#hostname ISP
ISP(config)#ipv6 unicast-routing
```

Konfiguracija Se0/0/1 sučelja

```
ISP(config)#interface serial 0/0/1
ISP(config-if)#ipv6 address FE80::C link-local
ISP(config-if)#ipv6 address 2001:DB8:CD1:C::1/64
ISP(config-if)#clock rate 128000
ISP(config-if)#no shutdown
```

Sada još treba napraviti *summery route*. Kako to napraviti? Prvo treba raspisati sve podmreže koje trebamo „sumirati"...

2001:DB8:DA:1::/64
2001:DB8:DA:2::/64
2001:DB8:DA:3::/64
2001:DB8:DA:4::/64
2001:DB8:DA:5::/64

...zapisati ih u dekomprimiranom obliku...

2001:0DB8:00DA:0001:0000:0000:0000:0000/64
2001:0DB8:00DA:0002:0000:0000:0000:0000/64
2001:0DB8:00DA:0003:0000:0000:0000:0000/64
2001:0DB8:00DA:0004:0000:0000:0000:0000/64
2001:0DB8:00DA:0005:0000:0000:0000:0000/64
...četvrte hekstete tj. hekstete koji definiraju podmreže napisati u binarnom obliku....

:0000000000000001
:0000 0000 0000 0010
:0000 0000 0000 0011
:0000 0000 0000 0100
:0000 0000 0000 0101

....i sada tražimo zadnji „zajednički bit“, a to je onaj označen crvenom bojom. Sada sve bite treba prebrojati od početka do uključujući crveni tj. zajedničkog bita (ima ih 61) i dobili smo „mrežni dio“ koji obuhvaća sve podmreže. 2001:DB8:DA::/61

Obzirom da se zajednički bit nalazi na poziciji koji označava broj 8, to znači da 2001:DB8:DA::/61 obuhvaća sve podmreže do broja 7 (ne samo naših 5). Sada to još moramo upisati u ISP usmjerivač

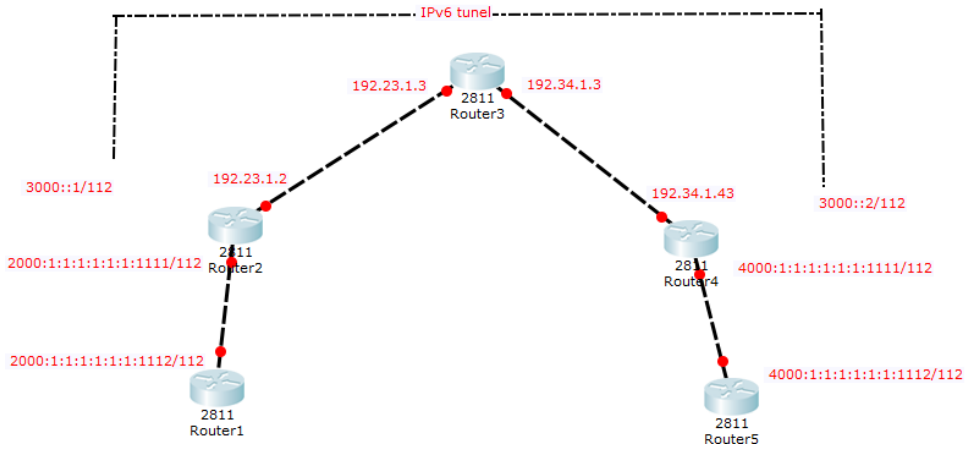
```
ISP(config)#ipv6 route 2001:DB8:DA::/61 s0/0/1
ISP#copy running-config startup-config
```

Upis statičke „*summary route*“

Spremanje trenutne konfiguracije usmjerivača u *startup* konfiguraciju

Zadatak 4 - IPv6 tuneliranje

Za zadanu mrežu uspostavi IPv6 tuneliranje



Slika 15.6. *Mrežna topologija sa adresama za zadatak 4*

Upute: Za ovu vježbu koristiti usmjerivače 2811 jer podržavaju IPv6.

Prvo je potrebno konfigurirati usmjerivače 1 i 5 koji predstavljaju čiste IPv6 mreže. Potrebno im je postaviti adrese sučelja, te aktivirati RIPng protokol za IPv6 mreže kao da IPv4 mreža ne postoji.

Usmjerivači 2 i 4 imaju sučelja na IPv4 i na IPv6 mreži. Na njima se vrši otvaranje virtualnog sučelja koje treba napraviti tunel kroz IPv4 mrežu. Na ovim usmjerivačima aktivira se i RIPng i OSPF protokol.

Usmjerivač 3 predstavlja čistu IPv4 mrežu i konfiguriramo ga kao da IPv6 mreža ne postoji

Konfiguracija Usmjerivača 1

```
Router(config)#hostname R1
R1(config)#ipv6 unicast-routing
R1(config)#interface f0/0
R1(config-if)#ipv6 address 2000:1:1:1:1:1:1:1112/112
R1(config-if)#no shutdown
R1(config-if)#ipv6 rip RIP1 enable
```

Konfiguracija Usmjerivača 5

```
Router(config)#hostname R5
R5(config)#ipv6 unicast-routing
R5(config)#interface f0/1
R5(config-if)#ipv6 address 4000:1:1:1:1:1:1:1112/112
R5(config-if)#no shut
R5(config-if)#ipv6 rip RIP1 enable
```

Konfiguracija Usmjerivača 2

```
Router(config)#hostname R2
R2(config)#ipv6 unicast-routing
R2(config)#interface f0/0
```

```
R2(config-if)#ipv6 address 2000:1:1:1:1:1:1:1111/112
R2(config-if)#ipv6 rip RIP1 enable
R2(config-if)#no shutdown
R2(config-if)#exit

R2(config)#interface f0/1
R2(config-if)#ip address 192.23.1.2 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#exit

R2(config)#interface tunnel0
R2(config-if)#ipv6 address 3000::1/112
R2(config-if)#ipv6 rip RIP1 enable
R2(config-if)#tunnel source f0/1
R2(config-if)#tunnel destination 192.34.1.4
R2(config-if)#tunnel mode ipv6ip
R2(config-if)#exit

R2(config)#router ospf 1
R2(config-router)#network 192.23.1.0 0.0.0.255 area 0
R2(config-router)#exit
```

Konfiguracija Usmjerivača 4

```
Router(config)#hostname R4
R4(config)#ipv6 unicast-routing
R4(config)#interface f0/0
R4(config-if)#ip address 192.34.1.4 255.255.255.0
R4(config-if)#no shutdown
R4(config-if)#exit

R4(config)#interface f 0/1
R4(config-if)#ipv6 address 4000:1:1:1:1:1:1:1111/112
R4(config-if)#ipv6 rip RIP1 enable
R4(config-if)#no shutdown
R4(config-if)#exit

R4(config)#interface tunnel0
R4(config-if)#ipv6 address 3000::2/112
R4(config-if)#ipv6 rip RIP1 enable
R4(config-if)#tunnel source f0/0
R4(config-if)#tunnel destination 192.23.1.2
R4(config-if)#tunnel mode ipv6ip
R4(config-if)#exit

R4(config)#router ospf 1
R4(config-router)#network 192.34.1.0 0.0.0.255 area 0
R4(config-router)#exit
```

Konfiguracija Usmjerivača 3

```
Router(config)#hostname R3
R3(config)#interface f0/1
R3(config-if)#ip address 192.23.1.3 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#exit

R3(config)#interface f0/0
R3(config-if)#ip address 192.34.1.3 255.255.255.0
```

```
R3(config-if)#no shutdown
R3(config-if)#exit

R3(config)#router ospf 1
R3(config-router)#network 192.23.1.0 0.0.0.255 area 0
R3(config-router)#network 192.34.1.0 0.0.0.255 area 0
R3(config-router)#exit
```

Literatura:

[1]Point-to-Point Protocol, CCERT-PUBDOC-2007-03-186,

https://www.cis.hr/www.edicija/LinkedDocuments/CCERT-PUBDOC-2007-03-186.pdf

(pristupljeno 09.03.2018.)

[2] IPv6 Protokol, CCERT-PUBDOC-2006-11-173.pdf

https://www.cis.hr/www.edicija/index.html (pristupljeno 10.10.2018.)

[3] VLAN Basic Concepts Explained with Examples,

https://www.computernetworkingnotes.com/ (pristupljeno 23.11.2017.)

[4] Računalne mreže – određivanje podmreža (subnetiranje)

https://sysportal.carnet.hr/node/443 (pristupljeno 12.10.2017.)

[5] IPsec VPN tunnel on Cisco routers using the Cisco IOS CLI. CCNA security topic.

http://danscourses.com (pristupljeno 15.12.2019.)

[6] Packet Tracer 8.2 tutorial - DHCP configuration,

https://www.packettracernetwork.com/tutorials/dhcpconfiguration.html#google_vignette

(pristupljeno 11.11.2018.)

[7] Determine What Impacts GRE Tunnel Interface States

https://www.cisco.com/c/en/us/support/docs/ip/generic-routing-encapsulation-gre/118361-technote-gre-00.pdf (pristupljeno 20.04. 2019.)