

## Podela IP adresa na manje mreže (subnetting)

Da bi se fundamentalno razumeo subnetting, potrebno je poznavati osnove IP adresiranja. Za to je najbolje konsultovati RFC791<sup>1</sup>.

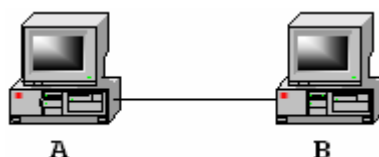
### Klasne mreže i namena netmaski

Originalni dizajn IP adresa prepoznaje nekoliko različitih klasa IP adresa. Ove klase razlikuju se kako po svojoj veličini (A, B, C), tako i po svojoj nameni (D, E). Za potrebe subnettinga, potrebno je razumeti razlike u veličini. Dakle, nećemo se baviti drugim razlikama između IP adresa.

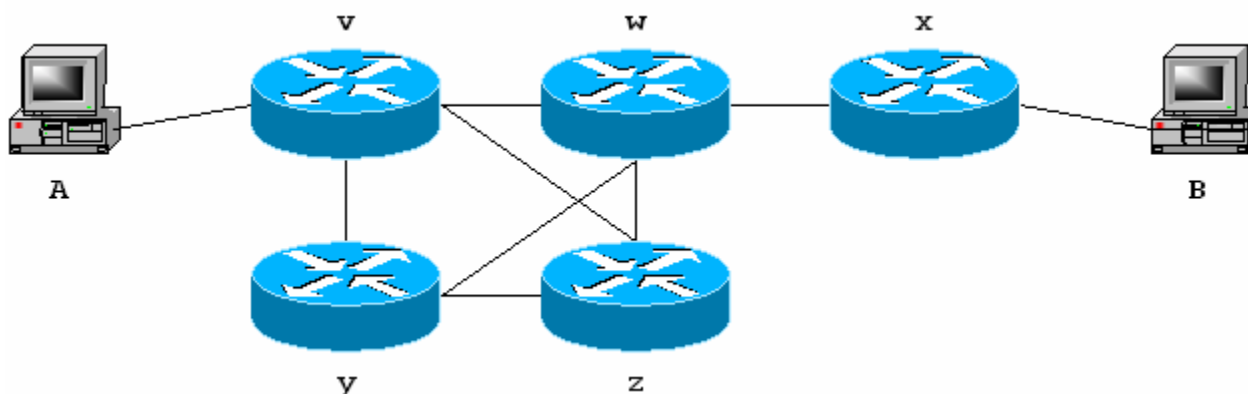
Osnovne klase IP adresa su:

- A. Osnovna maska: 255.0.0.0 (8 bita), opseg: 1.0.0.0 do 126.0.0.0, broj mreža:  $2^7 - 2$
- B. Osnovna maska: 255.255.0.0 (16 bita), opseg: 128.1.0.0 do 191.254.0.0, broj mreža:  $2^{14} - 2$
- C. Osnovna maska: 255.255.255.0 (24 bita), opseg: 191.0.1.0 do 223.255.254.0, broj mreža:  $2^{21} - 2$

Prva stvar koju se čovek zapita kad vidi netmasku je: "čemu ovo zapravo služi?". Da bi se odgovorilo na ovo pitanje, problem se mora sagledati van konteksta samih IP adresa. Poenta IP adresiranja je da se omogući komunikacija između dva uređaja na (IP - Internet) mreži. Da bi ta dva uređaja mogla međusobno da komuniciraju, moraju ne samo da poznaju adresu onog drugog, već da znaju putanju (rutu - route) do odredišta. Primer ispod je jednostavan:



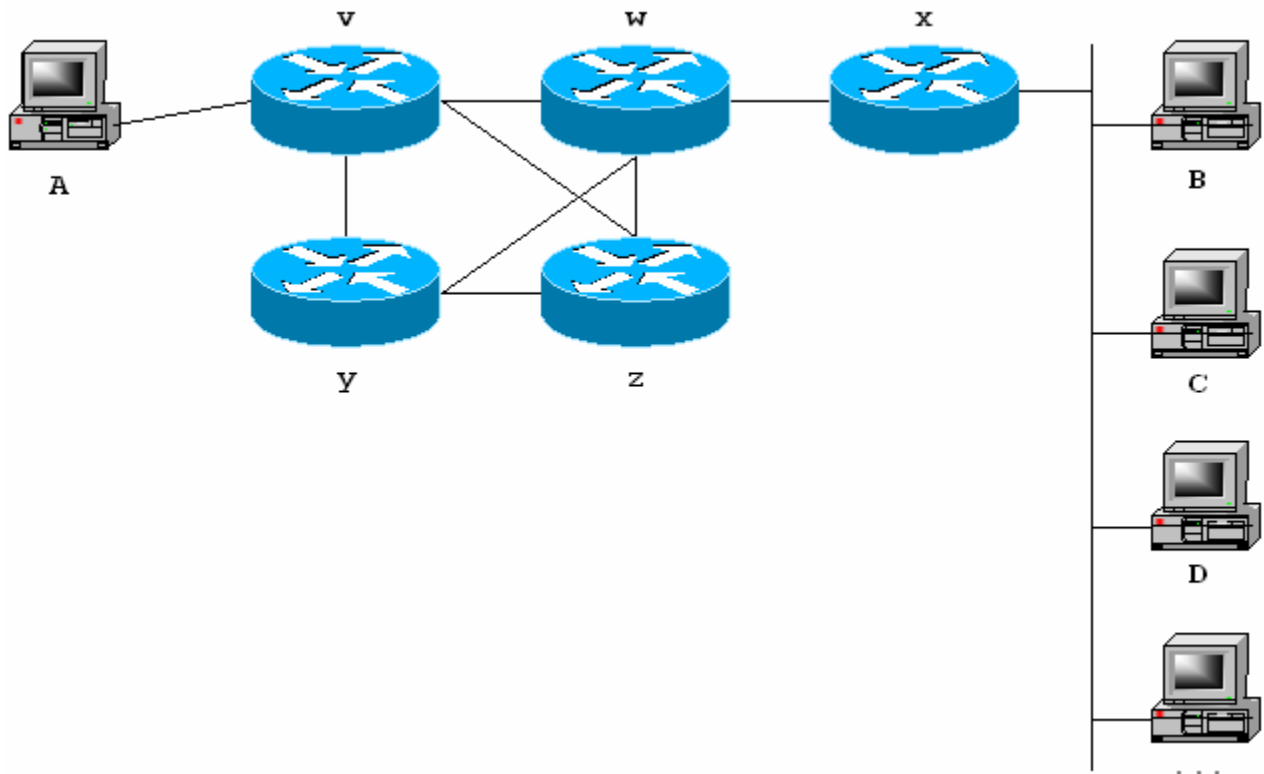
Računar A je direktno vezan sa računatom B. Problem sa znanjem putanje gotovo da ne postoji. Međutim, stvari u realnosti nisu tako jednostavne:



U primeru iznad, ne samo da A nema direktnu vezu sa B, već postoji više od nekoliko mogućih putanja od A do B. Svaka od imenovanih tačaka na dijagramu ima najmanje po jednu IP adresu. Da bi

<sup>1</sup> <ftp://ftp.ripe.net/rfc/rfc791.txt>

podaci od A stigli do B, da li A mora da zna punu putanju do B ili je dovoljno da zna putanju do [v]? U konkretnom slučaju, problem je jednostavan. Znatno kompleksniji problem se predstavlja pred, recimo [v]. Gde će on da pošalje paket? Iako bi očigledno bilo da je odredište [w], to zavisi od mnogo faktora. Da se ovo ne bi pretvorilo u veliko razmatranje o rutiranju, pretpostavimo da je putanja nebitna. Pogledajmo primer ispod:



Dakle, iza [x] imamo ne samo B, već C, D, E, F... Ako uzmemo u obzir da je mogući broj IP adresa preko 2000000000 i da su sve 32bitne, ako bi svaki uređaj na mreži znao putanju do svih adresa, dolazimo do problema da uređaji moraju da imaju stravično mnogo memorije da bi imali sve te informacije. Posebno što pored same adrese moraju imati i podatke o putanji. Kako rešiti ovaj problem?

Odgovor je netmaskama. Netmaske su način da se na jednostavan način referencira više IP adresa. Kako netmaske funkcionišu? Pogledajmo primer jedne IP adrese: *212.54.211.1*

Predstavljena binarno, ova adresa je: **11010100.00110110.11010011.00000001**

Predpostavimo da je to IP adresa našeg računara [B] u gornjem primeru. Neka su [C], [D] ... svi uvećani za 1. Dakle, imamo:

*B: 212.54.211.1*

*C: 212.54.211.2*

*D: 212.54.211.3*

...

Z: 212.54.211.254

Kako je ova IP adresa iz klase C (vidi tabelu sa klasama iznad), osnovna maska je: 255.255.255.0. Maska predstavljena binarno je:

```
11111111.11111111.11111111.00000000
```

Ako uradimo logičko "I" adrese B i maske:

```
11010100.00110110.11010011.00000001 (212.54.211.1)
11111111.11111111.11111111.00000000 (255.255.255.0)
-----
11010100.00110110.11010011.00000000 (212.54.211.0)
```

Ponovimo isto i za C:

```
11010100.00110110.11010011.00000010 (212.54.211.2)
11111111.11111111.11111111.00000000 (255.255.255.0)
-----
11010100.00110110.11010011.00000000 (212.54.211.0)
```

ili pak Z:

```
11010100.00110110.11010011.11111110 (212.54.211.254)
11111111.11111111.11111111.00000000 (255.255.255.0)
-----
11010100.00110110.11010011.00000000 (212.54.211.0)
```

U sva tri slučaja, dobijamo isti rezultat! Dati rezultat je najveća zajednička adresa za sve računare od A-Z. Ova adresa se zove "network" adresa. Dakle, u primeru nase mreže iznad, računar A (kao i svi ostali uređaji na putanji) ne mora da zna putanju do svih računara B-Z, već samo do mrežne adrese svih njih.

Network adresa je jedna od dve specijalne adrese u IP svetu. Druga specijalna adresa je tzv. "broadcast" adresa. Za razliku od mrežne adrese koja ima sve binarne 0 na kraju (host polje IP adrese), broadcast adresa ima sve binarne 1 u host polju. Dakle, broadcast adresa za naš primer je 212.54.211.255. Namena broadcast adrese prevazilazi namenu ove poruke.

Pogledom na masku iznad vidimo da njom "pokrivamo" 256 adresa. Šta ukoliko naša mreža ima manje računara? Ovde dolazimo do osnovnog problema kojim se ova poruka bavi - subnettingom. Subnetting (neki nakaradan prevod na naš jezik bi mogao da bude podmrežavanje) je proces kojim se iz default "host polja" pozajmljuje određeni broj bitova i dodaje maske, kako bi se mreža podelila na manje funkcionalne (administrativne) celine.

Recimo da se u našem primeru iznad, krajnja mreža sastoji od pedeset računara sa adresama 212.54.211.1 - 212.54.211.50. U tom slučaju, preko 200 adresa sa default maskom bi bilo neupotrebljeno. Ovaj problem se može rešiti tako što bi se mreža podelila u subnete.

Najmanji subnet koji može da opsluži 50 računara je od 64 adrese ( $2^6$ ). Iznad sam spomenuo da je subnetting proces pozajmljivanja bitova iz host polja i dodavanja maske. Dakle, da bismo imali subnet od 64 adrese, nama je potrebno 6 umesto 8 bitova za host. Dakle, maske treba dodati 2 bita. Pogledajmo kako to izgleda.

Network adresa:

212.54.211.0

11010100.00110110.11010011.{00}000000 (212.54.211.0)

11111111.11111111.11111111.{11}000000 (255.255.255.192)

Sa {} su označeni pozajmljeni bitovi. Ovo polje bitova se naziva "subnet polje".

Da bi naša mreža ispravno funkcionisala, svaki računar na mreži je potrebno da ima konfigurisanu netmasku iznad. Promena netmaske sa default maske nam stvara još jedan problem. Broadcast adresa više nije ista! Da podsetim, broadcast adresa u host polju ima sve binarne 1. Pozajmljivanjem bitova za subnet polje, host polje je sad 6 bitova. Dakle, broadcast adresa je:

11010100.00110110.11010011.{00}111111 (212.54.211.63)

Uz malo sreže ovo je do sada jasno. Sad sledi zaplet. Po klasnim pravilima subnettinga, ovaj primer iznad je **NEISPRAVAN!** Ovo je veoma specifičan problem vezan za rutiranje. Naime, network adresa našeg subneta (212.54.211.0) je ista kao i adresa C klase 212.54.211.0. Ovo nije dozvoljena situacija. Još jedna nedozvoljena situacija je pravljenje subneta koji u sebi sadrži default broadcast adresu mreže (212.54.211.255). Ovo treba uzeti u obzir kao aksiom za klasne mreže!<sup>2</sup>

Veoma često praktično pitanje je, kako za datu adresu i netmasku odrediti koje su network i broadcast adresa. Ovaj problem ću prvo da objasnim binarno, a onda da otkrijem trik za veoma brzo računanje bez konverzije u binarni sistem (na kraju teksta).

Primer:

---

<sup>2</sup> Ako su se korisnici Cisco rutera zapitali zašto im ne radi "default route", a problem se reši komandom *ip classless*, objašnjenje počinje od ovog aksioma.

174.211.141.177  
255.255.224.0

10101110.11010011.10001101.10110001 (174.211.141.177)  
11111111.11111111.11100000.00000000 (255.255.224.0)

Logičkim "1" dobijamo adresu mreže:

10101110.11010011.10000000.00000000 (174.211.128.0)  
11111111.11111111.11100000.00000000 (255.255.224.0)

Ako u host polje (sva polja koja su u netmasci 0) stavimo sve 1, dobijamo broadcast adresu:

10101110.11010011.10011111.11111111 (174.211.159.255)

### **Bezklasna pravila subnettinga**

Kako je vreme prolazilo, tako je Internet rastao. Sve više firmi se povezivalo na mrežu. Mnoge od organizacija koje su se povezivale na Internet, imale su manje računara nego sto su klasne granice (C:  $2^8 - 2$ , B:  $2^{16} - 2$ , A:  $2^{24} - 2$ ). Subnetting je rešavao ovaj problem samo do izvesne granice. Zašto? Osnovni problem klasnog subnettinga je što ne dozvoljava subnete različite veličine. Na primer, mrežu B klase: 142.14.0.0 je bilo moguće podeliti, recimo na 254 ( $2^8 - 2$ ) mreže: 142.14.1.0, 142.14.2.0 ... 142.14.254.0 sa netmaskom 255.255.255.0. Pretpostavimo da je data B klasa bila dodeljena Internet provajderu kojem se javio korisnik koji u firmi ima samo 2 računara, a želi da se poveže na Internet. Provajder nije imao šta drugo da radi, već da tom korisniku dodeli čitav subnet od 256 adresa. Ako uzmemo u obzir da je korisnik od ovoga mogao da koristi samo 254, dakle 251 adresa je neiskorišćena!

Ubrzo je postalo jasno da Internetu predstoji ozbiljan problem - nedostatak IP adresa. Posebno IP adresa iz B klase.

Rešenje problema se javilo uz pomoć dve tehnologije (pravilnije bi bilo reći dve teorije). Obe teorije bave se ukidanjem klasa IP adresa. Prva teorija je VLSM (Variable Length Subnet Mask - RFC1878<sup>3</sup>), a druga CIDR (Classless Inter Domain Routing - RFC1519<sup>4</sup>).

Ukratko (za detalje pogledajte dva navedena RFC-a), ova dva RFC-a se bave mogućnošću da se se omogući deljenje kompletnog IP opsega (0.0.0.0 - 255.255.255.255) na različite delove, ne po klasama, već provizorno. Dakle, ukidaju se "podklase" i čitav adresni prostor se smatra jednom mrežom, koja se može deliti na "provizorno" male delove, različite veličine (dužine). Ovime se bavi VLSM. CIDR je koncept koji omogućava uređajima na mreži (uglavnom ruterima) da međusobno razmenjuju informacije o putanjama (rutama) u mreži pritom imajući u vidu izmenjena pravila.

---

<sup>3</sup> <ftp://ftp.ripe.net/rfc/rfc1878.txt>

<sup>4</sup> <ftp://ftp.ripe.net/rfc/rfc1519.txt>

Sa stanovišta subnetovanja, bitno je shvatiti par osnovnih razlika u odnosu na klasna pravila:

- Odmah zaboraviti klase adresa. One ne postoje. Postoji samo jedna "klasa" - kompletan adresni prostor!
- IP adresa je jedan 32bitni broj - ne četiri 8bitna broja.
- Ogranicenje za "mrežni" i "broadcast" subnet postoje i dalje. IP adrese iz subneta 0.0.0.0/255.0.0.0 i 255.0.0.0/255.0.0.0 su neispravne.

Iako nije netačno, kao netmaska se više ne preporučuje upotreba tzv. "dotted decimal" formata (npr. 255.255.255.0), već tzv. "prefix" format (/24). Prefix format nije nista drugo do informacija o tome koliko "konstantih" bitova ima u masci. Konstantni bitovi su bitovi postavljeni na 1. Tabela ispod je brz konvertor iz "prefix" u "dotted decimal" zapis:

Prefix	Dotted-decimal	Veličina
/32	255.255.255.255	1
/31	255.255.255.254	2
/30	255.255.255.252	4
/29	255.255.255.248	8
/28	255.255.255.240	16
/27	255.255.255.224	32
/26	255.255.255.192	64
/25	255.255.255.128	128
/24	255.255.255.0	256
/23	255.255.254.0	512
/22	255.255.252.0	1024
/21	255.255.248.0	2048
/20	255.255.240.0	4096
/19	255.255.224.0	8192
/18	255.255.192.0	16384
/17	255.255.128.0	32768

Prefix	Dotted-decimal	Veličina
/16	255.255.0.0	65536
/15	255.254.0.0	131072
/14	255.252.0.0	262144
/13	255.248.0.0	524288
/12	255.240.0.0	1048576
/11	255.224.0.0	2097152
/10	255.192.0.0	4194304
/9	255.128.0.0	8388608
/8	255.0.0.0	16777216
/7	254.0.0.0	33554432
/6	252.0.0.0	67108864
/5	248.0.0.0	134217728
/4	240.0.0.0	268435456
/3	224.0.0.0	536870912
/2	192.0.0.0	1073741824
/1	128.0.0.0	2147483648

Iako je na prvi pogled novi princip subnetovanja jednostavniji od klasnog (i jeste!), mnogima koncept bezklasnosti predstavlja problem. Razlog za ovo je klasna terminologija koja je još uvek veoma u upotrebi. Većina i dalje blok IP adresa od 256 adresa naziva "C klasom". Mnogi ljudi ovo rade bez obzira na to što mnoge adrese zapravo originalno ni ne bi bile C klasa, već subnet neke od B klasa (recimo: 147.91.75.0/24). Drugi koncept koji se javlja je mogućnost da se uz pomoć prefix notacije referencira veći adresni prostor od jedne klase, a da se pritom koriste adrese koje očigledno pripadaju toj klasi. Primer: 194.250.64.0/23. Da bi se olakšao prelaz sa klasne na bezklasnu terminologiju, osmišljen je naziv "supernet" koji označava datu situaciju. Praktično posmatrano, i subnet i supernet su nepotrebni izrazi. Najispravnije je koristiti izraz blok.

Generalna pravila za subnetovanje i dalje postoje. Svaki blok mora da počne od adrese koja za dat prefix (netmasku) u host polju ima sve binarne 0 i da se završi na adresi koja u host polju ima sve 1. No, ne zaboravimo da se ceo adresni prostor tretira kao jednak - nema klasa!

Evo primera.

Od provajdera je dobijen blok IP adresa: 62.11.48.0/23. Potrebno je napraviti dva subneta za 50 računara, jedan za 200, a ostatak prostora iskoristiti za point-to-point veze (ruter-ruter, dakle potrebne po dve adrese).

Za početak, kako dodeljeni blok izgleda binarno:

```
001111110000010110011000{000000000}
111111111111111111111111{000000000}
```

Obratite pažnju da nisam koristio "." da razdvojim adresu i masku na oktete. To je zbog lakšeg razumevanja. Takođe, obratite pažnju da je host deo (označen sa { }) 9 bita. Pogledom na tabelu iznad, dobijamo da prefix /23 omogućava ukupno 512 adresa.  $2^9=512$ . Za sad smo na dobrom putu. Naš problem kaže da nam je potreban jedan subnet za 200 računara. Opet pogledom na tabelu, vidimo da jedini prefix koji nam odgovara je /24. Ako isto ponovimo za dva subneta od po 50 računara, vidimo da je najbliži prefix /26. Za 4 adrese, prefix je /30<sup>5</sup>. Dakle, da bismo subnetovali nas blok, trebaju nam sledeci blokovi:

```
/24
/26
/26
/30
/30
...
```

Jedan koristan savet: Iako za to nema specijalnog razloga, obično je lakše izvršiti subnetovanje određenog adresnog prostora ukoliko se krene od najvećeg ka najmanjem bloku. *Ovo nije pravilo!*

Dakle, imajući u vidu date prefixe, kao i savet iznad, evo šta dobijamo kao rezultat našeg malog problema (ovo sto sledi je klasično subnetovanje objašnjeno u prethodnom tekstu o klasnom subnetovanju - neću ponavljati objašnjenja):

Blok za 200 adresa:

```
00111111000001011001100000000000 62.11.48.0
111111111111111111111111100000000 /24
```

Ostaje:

```
001111110000010110011000100000000 62.11.49.0
111111111111111111111111100000000 /24
```

<sup>5</sup> U poslednje vreme, mogućnost korišćenja /31 prefixa za point-to-point veze se polako uvodi u upotrebu. Ovo još uvek nije široko rašireno.

Odavde su nam je potrebna dva /26 bloka:

```
00111110000010110011000100000000 62.11.49.0
11111111111111111111111111111111000000 /26
```

```
00111110000010110011000101000000 62.11.49.64
11111111111111111111111111111111000000 /26
```

Preostaju nam /30 blokovi. Evo dva sledeća:

```
00111110000010110011000110000000 62.11.48.128
11111111111111111111111111111111100 /30
```

```
00111110000010110011000110000100 62.11.48.132
11111111111111111111111111111111100 /30
```

...

### **Kako bez konverzije u binarni sistem**

Sav dosadašnji tekst je bazirao računice na konverziji u binarni sistem i razumevanju osnovne binarne logike i aritmetike (binarno sabiranje). To može predstavljati izvesan problem. Srećom, postoji i skraćena metoda za brzo računanje subneta (network i broadcast adresa) na osnovu date IP adrese i netmaske (ukoliko je dat prefix, poželjno je znati tabelu konverzije prefix-netmaska napamet).

Da ne bih previše zalazio u teoriju, evo primera:

```
216.221.4.12/255.255.224.0
```

Da bi se izračunala network adresa, potrebno je napisati IP adresu i netmasku jednu iznad druge. Gde se u masci nalazi "255", prepisati broj iz IP adrese. Na mestu gde je u netmasci 0, napisati 0.

```
216.221.4.12
255.255.224.0
-----
216.221.    .0
```

Ukoliko ne ostane praznina nigde, to je network adresa. Ukoliko ostane praznina, preostali broj u netmasci oduzeti od 256. Ovo je veličina mreže. U našem primeru, veličina mreže je 32. Nazovimo ovaj broj množiocem. Da bismo dobili ispravan subnet, potrebno je naći proizvod množioca koji je



najbliži preostalom broju u IP adresi, a da je manji od tog broja. U našem slučaju, ovo je 0. Dobijeni rezultat je network adresa:

```
216.221.4.12
255.255.224.0
-----
216.221.0.0
```

Kad smo izračunali network adresu, izračunavanje broadcast adrese je veoma jednostavno. Na prvi oktet u network adresi koju se nalazi iznad broja koji nije 255 u netmasci dodati množilac umanjen za jedan. Sve ostale 0 u network adresi zameniti sa 255. U konkretnom primeru:

```
216.221.4.12
255.255.224.0
-----
216.221.0.0
216.221.31.255
```

Evo još jednog, možda malo komplikovanijeg primera (korišćen algoritam iznad):

```
62.74.52.41      62.74.52.41      62.74.52.41      62.74.52.41
255.255.248.0 ==> 255.255.248.0 ==> 255.255.248.0 ==> 255.255.248.0
-----
62.74.  .0      [8] 62.74.48.0      62.74.48.0
62.74.55.255
```

U primeru iznad, množilac je 8. Proizvod množioca koji nam je potreban je broj koji se dobija umnožavanjem 8, a da je proizvod najbliži, a manji od 52. Taj broj je 48. Broadcast adresu smo dobili tako što smo na 48 dodali (8 - 1).