

Računalniške komunikacije

2023/24

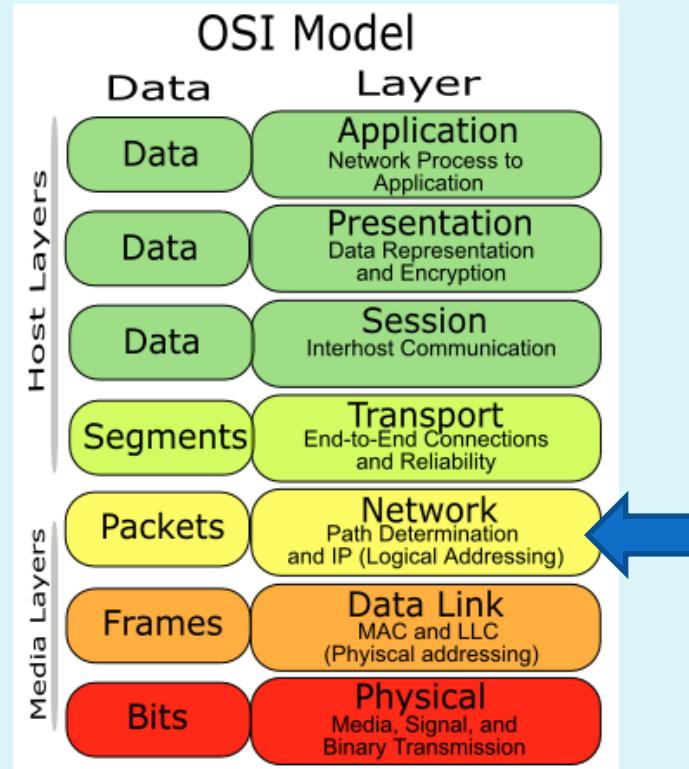
omrežna plast

storitve omrežne plasti, usmerjevalniki,
vrste omrežij, posredovanje,
IPv4, fragmentacija, naslavljanje,
podomrežja, hierarhija,
DHCP, NAT, ICMP,
IPv6, usmerjanje

Povzetek: povezavna plast

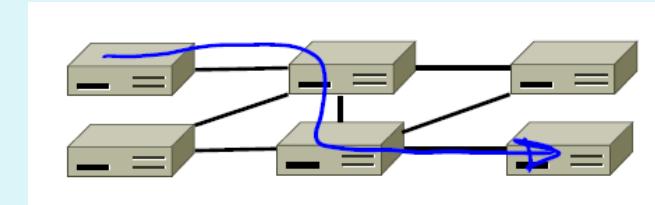
- naloge/storitve povezavne plasti:
 - **okvirjanje**: okvir, enkapsulacija/dekapsulacija po plasteh komunikacijskega modela
 - **zaznavanje in odpravljanje napak**: EDC, pariteta (1D/2D, soda/liha), Hammingova koda, CRC
 - **dostop do medija** s protokoli MAC
 - delitev kanala: TDMA, FDMA
 - naključni dostop: ALOHA, razsekana ALOHA, CSMA, CSMA/CD, CSMA/CA (RTS, CTS)
 - izmenični dostop: centralno vozlišče, krog z žetonom
 - **zanesljiva dostava, kontrola pretoka**
- **protokola**:
 - Ethernet: konceptualna zasnova, storitve protokola, topologije, struktura okvirja
 - PPP (point-to-point protocol): struktura okvirja, vrivanje podatkov
- **naslavljjanje na povezavni plasti** (48-bitni fizični naslovi MAC)
- aktivna oprema (ponavljalnik, razdelilnik, stikalo)
- delovanje **stikal**
 - stikalna tabela; akcije: **poplavljjanje, posredovanje, filtriranje**
 - vloga stikal v topologiji omrežja (medsebojno povezovanje stikal), VLAN

Omrežna plast



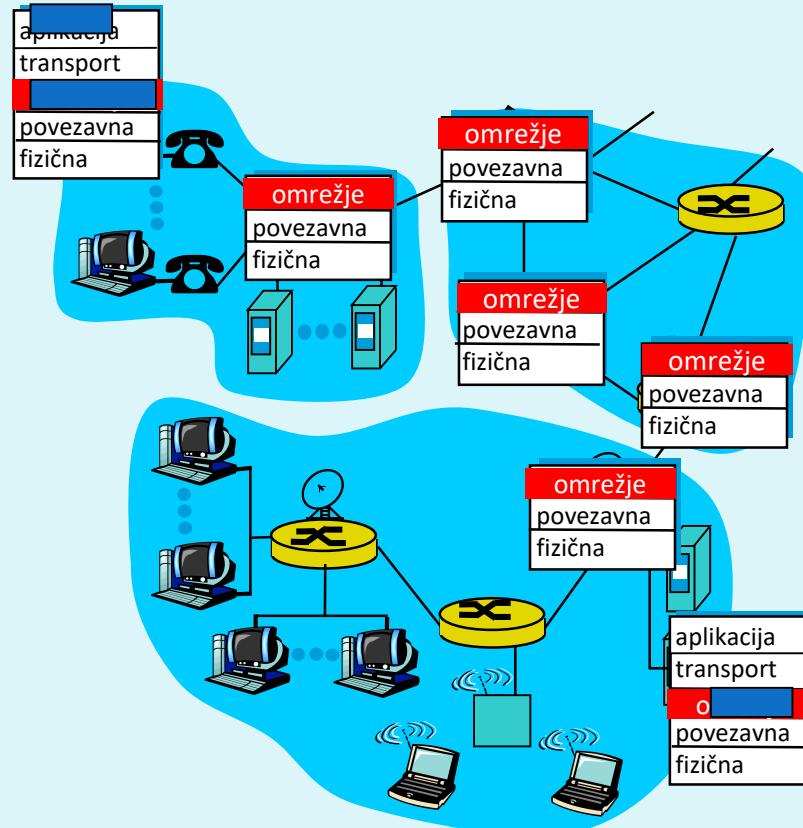
Naloge omrežne plasti:

- transport segmenta od **KONČNEGA** pošiljatelja do prejemnika
- enkapsulacija segmentov transportne plasti v **pakete** na strani pošiljatelja
- prisotna v vseh omrežnih napravah in v jedru omrežja (=usmerjevalnikih)

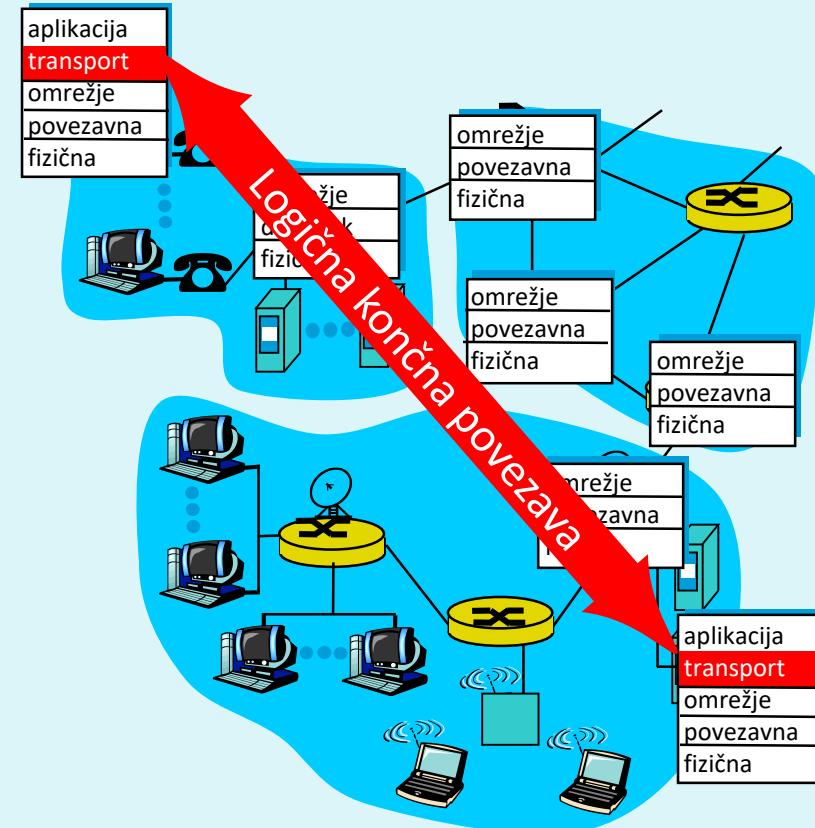


Naloga omrežne in transportne plasti

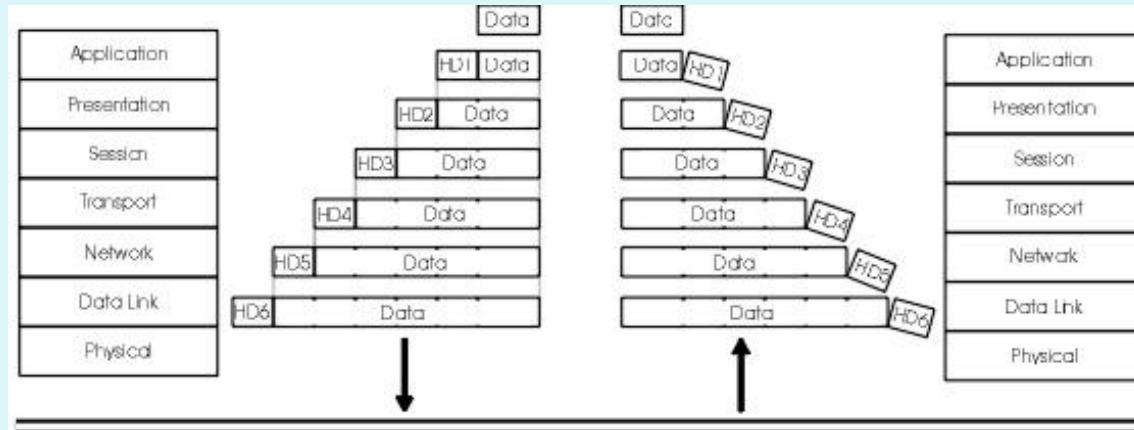
- Od računalnika do računalnika



- Od procesa do procesa

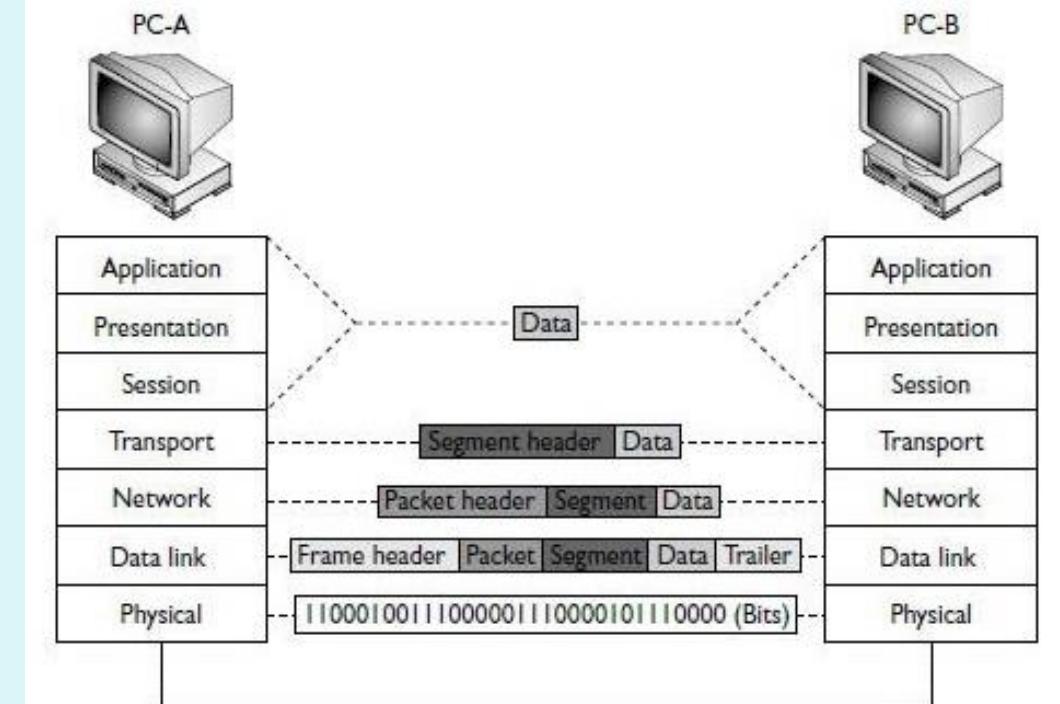


Enkapsulacija in dekapsulacija na omrežni plasti



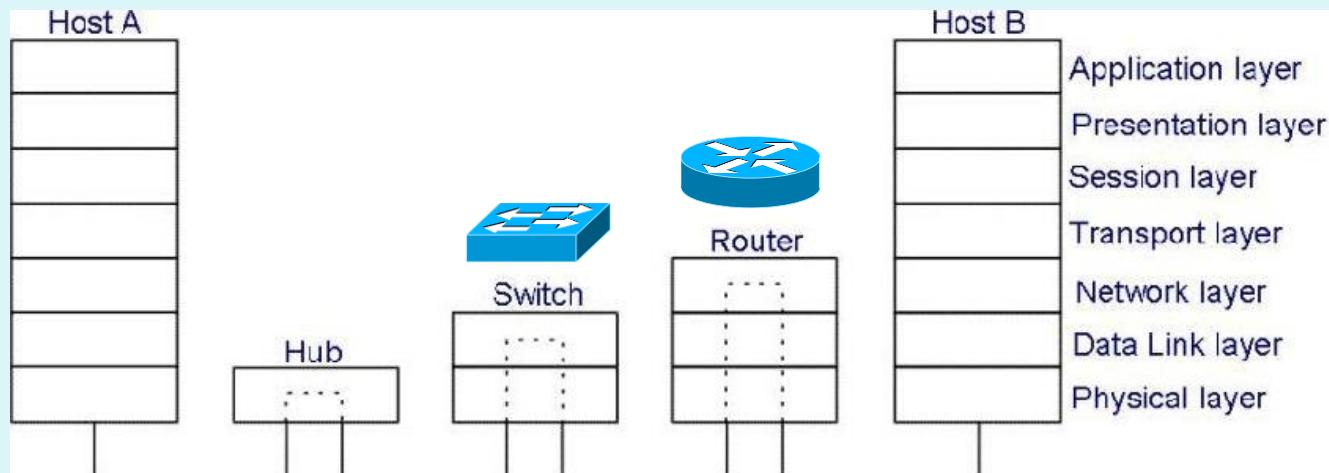
Podatki katere plasti se:

- enkapsulirajo?
- dekapsulirajo?



Usmerjevalnik

- naprava, ki deluje na omrežni plasti in skrbi za transport datagrama po jedru omrežja
 - povezave med različnimi mediji in protokoli
 - izvajajo **usmerjanje** in **posredovanje**

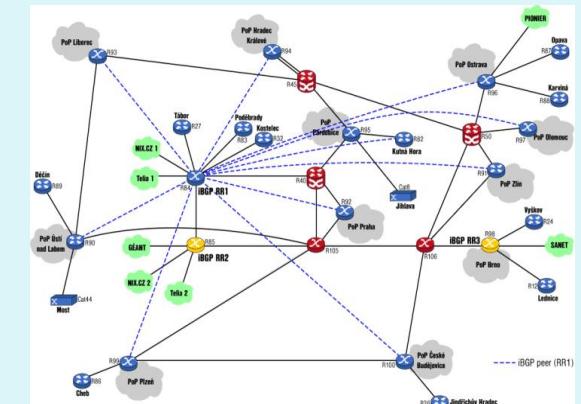
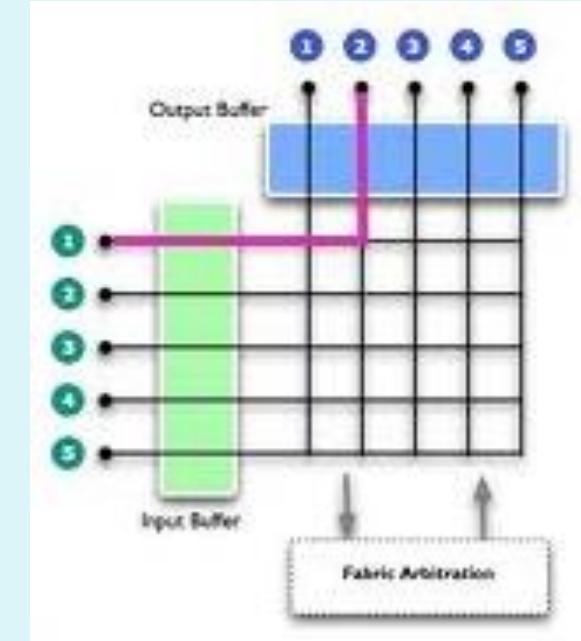


Application layer
Presentation layer
Session layer
Transport layer
Network layer
Data Link layer
Physical layer



Funkciji usmerjevalnika

- **posredovanje paketov (forwarding)**: prenos paketa iz vhodnega na izhodni vmesnik usmerjevalnika. Poteka znotraj posameznih usmerjevalnikov!
 - usmerjevalnik ima **posredovalno tabelo** (forwarding table) na podlagi katere določa, na katera izhodna vrata poslati paket
 - *analogija: določitev točke vstopa v posamezen kraj na poti in izstopa iz njega*
- **usmerjanje (routing)**: določitev poti paketov od izvora do cilja. Je "kolektivno delo" vseh omrežnih naprav na poti, ki izvajajo usmerjevalne algoritme (in protokole).
 - *analogija: planiranje poti od Loma pod Storžičem do Kopra*
 - izvajajo usmerjevalni protokoli



Storitve omrežne plasti

Omrežna plast *lahko omogoča* naslednje storitve:

1. **zagotovljena dostava** paketov
2. dostava paketov v **zagotovljenem času**
3. dostava paketov v **pravem zaporedju**
4. zagotovljena spodnja **meja pasovne širine**
5. največja dovoljena **varianca zakasnitve (jitter)**:

$$t_{\text{pošiljanja}}(P_2) - t_{\text{pošiljanja}}(P_1) \approx t_{\text{prejetja}}(P_2) - t_{\text{prejetja}}(P_1)$$

6. **varno komunikacijo** (zaupnost, integriteto podatkov, avtentikacijo)



Storitve Interneta

- Katere od naštetih storitev zagotavlja Internet?

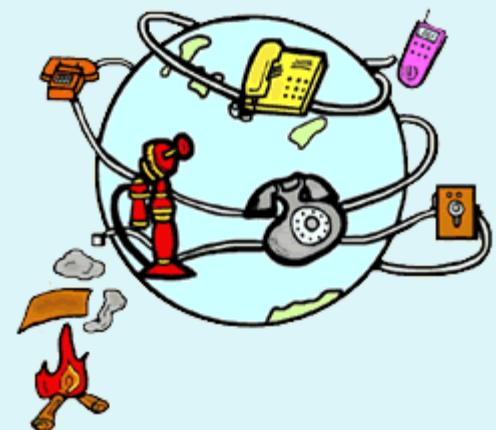
Prav nobene. ☺

(best-effort service)

"Best-effort service is a euphemism for no service at all"

Omrežje	Model	zagotovljene storitve				
		pas. širina	brez izgube	Vr. red	Čas	obv. o zamašitvi
Internet	best effort	ne	ne	ne	ne	ne (izguba)
ATM	CBR constant bit rate	konstantna	da	da	da	ni zamašitev
ATM	ABR available bit rate	minimalna	ne	da	ne	da

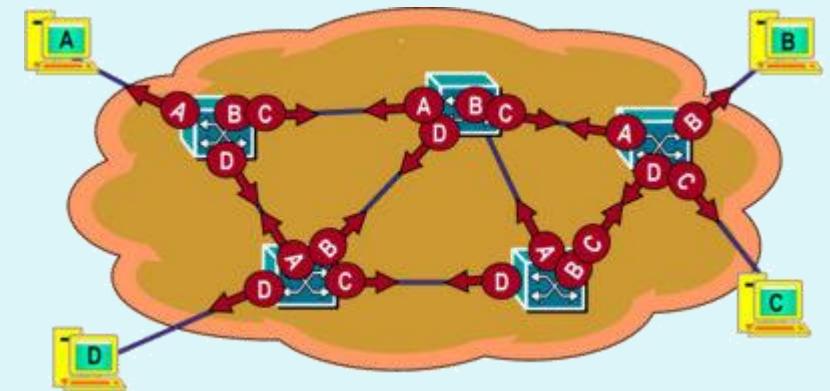
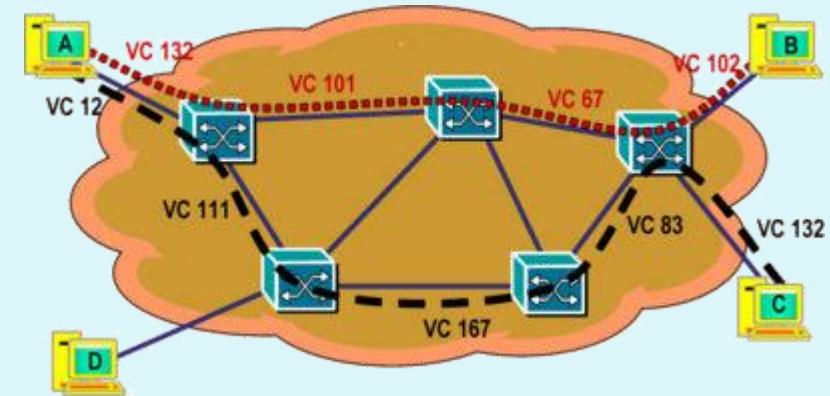
Povezavna in nepovezavna omrežja



Povezavna in nepovezavna omrežja

- **povezavna omrežja** (navidezni vodi) omogočajo vzpostavitev zveze v omrežni infrastrukturi med pošiljateljem in prejemnikom
 - **nepovezavna omrežja** (datagramska, paketna) omogočajo posredovanje paketov skozi infrastrukturo brez vzpostavljenih povezav

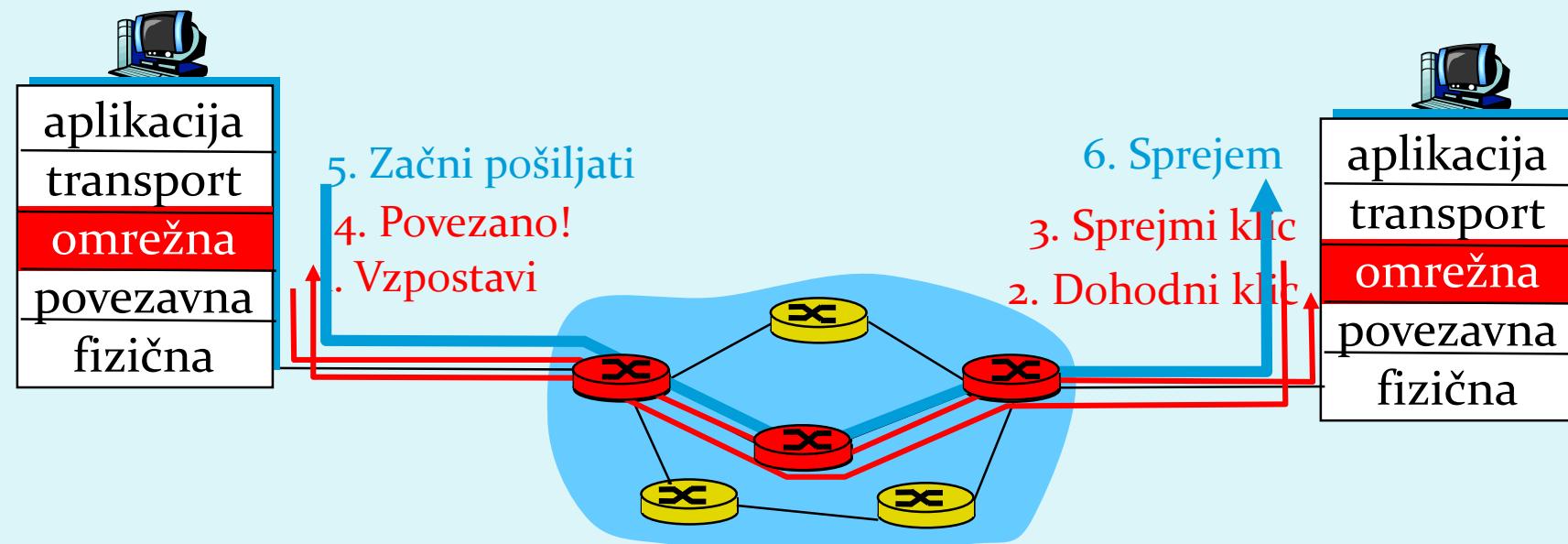
=> Kam spada Internet?



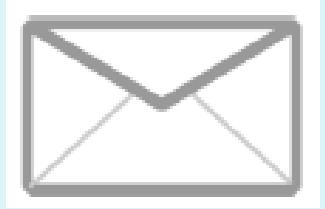
Navidezni vodi



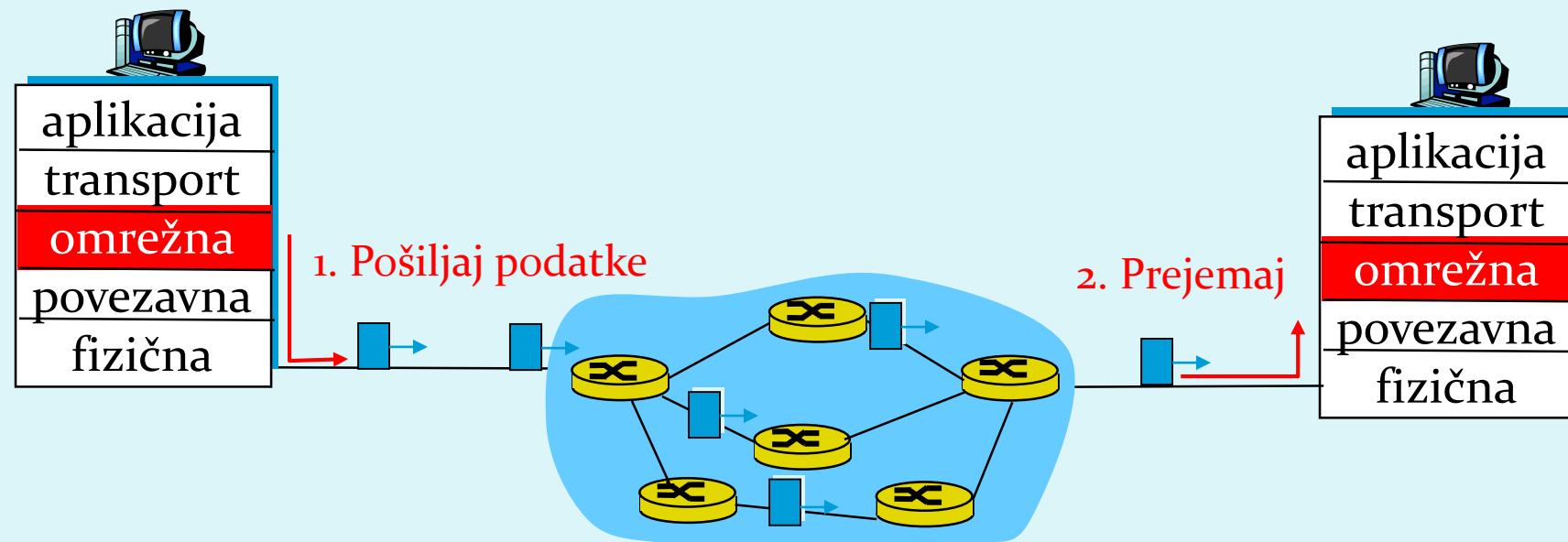
- podobno kot telefonske zveze
- faze pri izvedbi navideznega voda: vzpostavitev, tok podatkov, rušenje
- številke vodov na povezavah neodvisne, kar omogoča lažjo konfiguracijo
- usmerjevalniki usmerjajo pakete glede na **številke vodov**
- uporaba: ATM, X.25, MPLS, Frame Relay (ne Internet!)



Datagramska omrežja

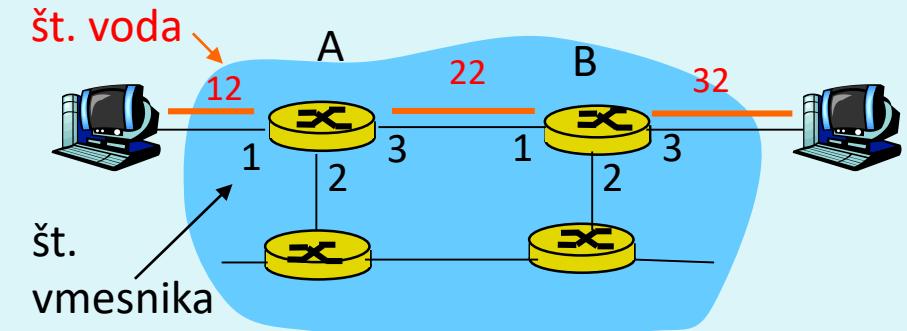


- ni faze vzpostavljanja povezave
- usmerjevalniki ne hranijo podatkov o končnih povezavah
- paketu se doda **naslov cilja** in se ga "vrže" v omrežje
- usmerjevalniki posredujejo **glede na ciljni naslov** v paketu
- paket lahko med istim izvorom in ciljem potuje po različnih poteh



Navidezni vodi: posredovalne tabele

- posredovalna tabela se nahaja v usmerjevalniku: vsebuje podatke za posredovanje paketov
- paketi so označeni z **identifikatorjem voda**



Vhodni vmesnik	Vhodna št. voda	Izhodni vmesnik	Izhodna št. voda
1	12	3	22
2	63	1	18
3	7	2	17
...



Vhodni vmesnik	Vhodna št. voda	Izhodni vmesnik	Izhodna št. voda
1	22	3	32
1	34	2	23
2	4	1	55
...



Datagramsko omrežje: posredovalne tabele

- uporabljamo 32-bitne naslove pošiljateljev in prejemnikov
- naslovimo lahko $2^{32}=4$ milijarde naslosov, kar bi zahtevalo ogromne posredovalne tabele
- **(možna) REŠITEV 1:** združimo dele nasloov v range (razpone nasloov):

Ciljni naslov	Vmesnik povezave
Od 11001000 00010111 00010000 00000000	0
Do 11001000 00010111 00010111 11111111	
Od 11001000 00010111 00011000 00000000	1
Do 11001000 00010111 00011000 11111111	
Od 11001000 00010111 00011001 00000000	2
Do 11001000 00010111 00011111 11111111	
sicer	3

- **REŠITEV 2:** posredujemo na podlagi PREDPONE (prefiksa) - začetnih bitov naslova ← to je zanimivo!

Ujemanje najdaljše predpone (longest prefix match)

- namesto pisanja rangov:

Ciljni naslov	Vmesnik povezave
Od 11001000 00010111 00010000 00000000 Do 11001000 00010111 00010111 11111111	0
Od 11001000 00010111 00011000 00000000 Do 11001000 00010111 00011000 11111111	1
Od 11001000 00010111 00011000 00000000 Do 11001000 00010111 00011111 11111111	2
sicer	3

definiramo usmerjanje na podlagi predpon, torej:

Ciljni naslov	Vmesnik povezave
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
sicer	3

Ujemanje najdaljše predpone (longest prefix match)

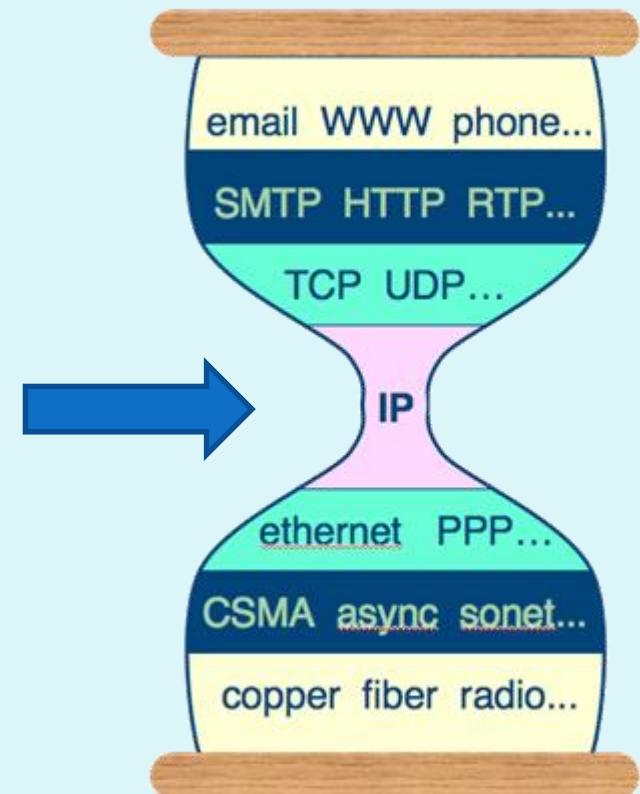
Ciljni naslov	Vmesnik povezave
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
sicer	3

- zapis s predponami je krajši in bolj učinkovit
- če ustreza več predpon, uporabimo ujemanje NAJDALJŠE. Primer:
naslov 11001000 00010111 00011000 10100001 se ujema z naslednjima zapisoma:
 - ❖ 11001000 00010111 00011000 10100001 (vmesnik 1)
 - ❖ 11001000 00010111 00011000 10100001 (vmesnik 2)
 - ❖ usmerimo ga na vmesnik 1 (najdaljše ujemanje)
- usmerjevalniki morajo hraniti posredovalne tabele in stanje o povezavah
 - za njihovo avtomatsko posodabljanje skrbijo **usmerjevalni algoritmi** (njihovo delo je počasno - v intervalu nekaj sekund - v primerjavi z vzpostavitvenim časom navideznih vodov - nekaj mikrosekund)

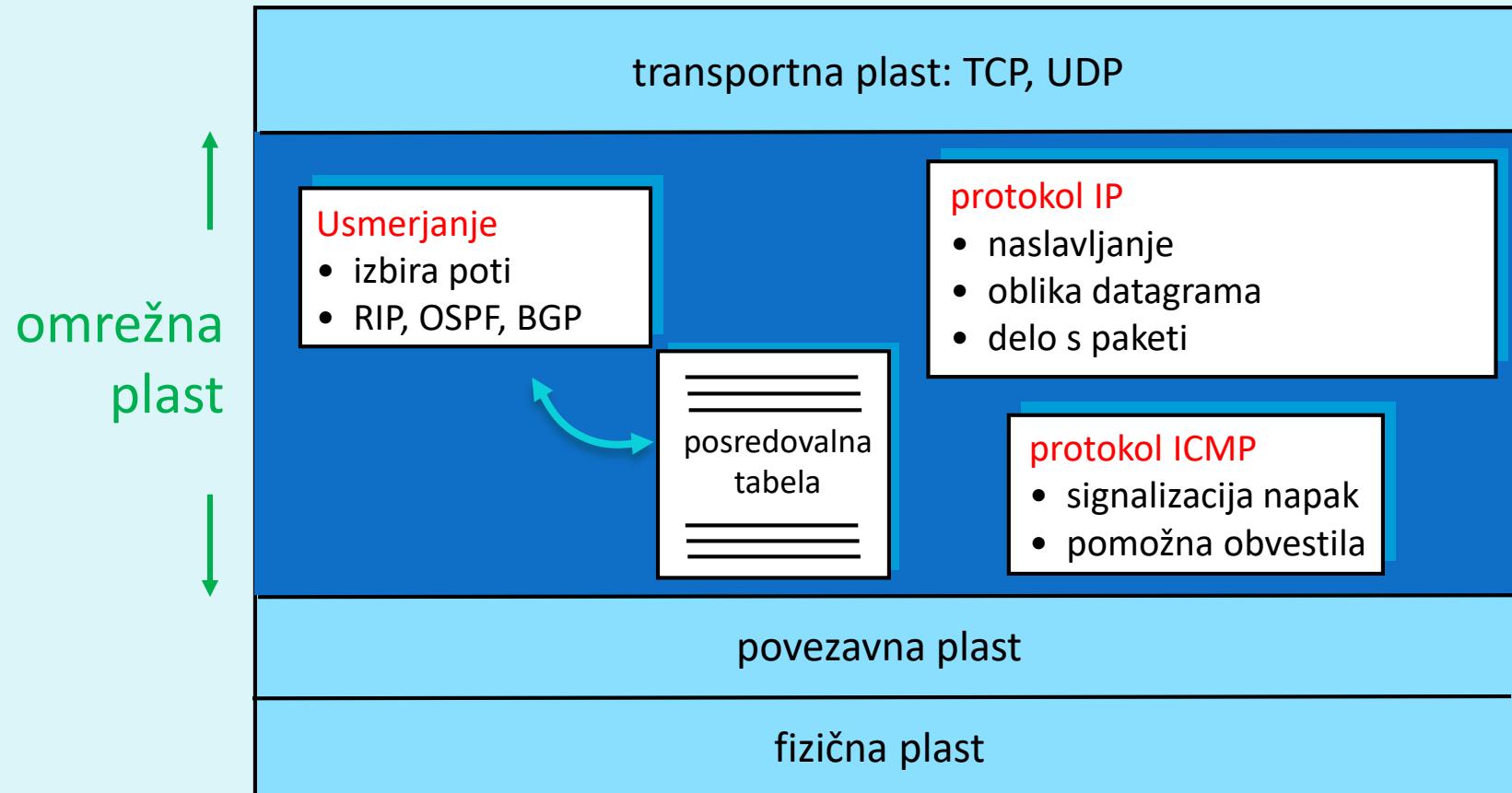
Primerjava obeh tipov omrežij

Internet (datagramsko)	ATM (VC omrežje)
usmerjanje glede na ciljni naslov	usmerjanje glede na ID voda
komunikacija med računalniki : zato so dovoljene elastične storitve, kjer čas ni tako pomemben	izvira iz telefonije : zakasnitev in zanesljivost sta pomembna
težka zagotovila kakovosti	preprosta zagotovila kakovosti
<ul style="list-style-type: none">• končni sistemi so "pametni", znajo sami popravljati napake in izvajati manjkajoče storitve• omrežje je preprosto	<ul style="list-style-type: none">• končni sistemi so "neumni"• omrežje je kompleksno, saj mora zagotavljati storitve kakovosti
preprosto dodajanje novih storitev (aplikacij) in povezovanje heterogenih omrežij	težje dodajanje novih storitev, pogojeno z infrastrukturo omrežja

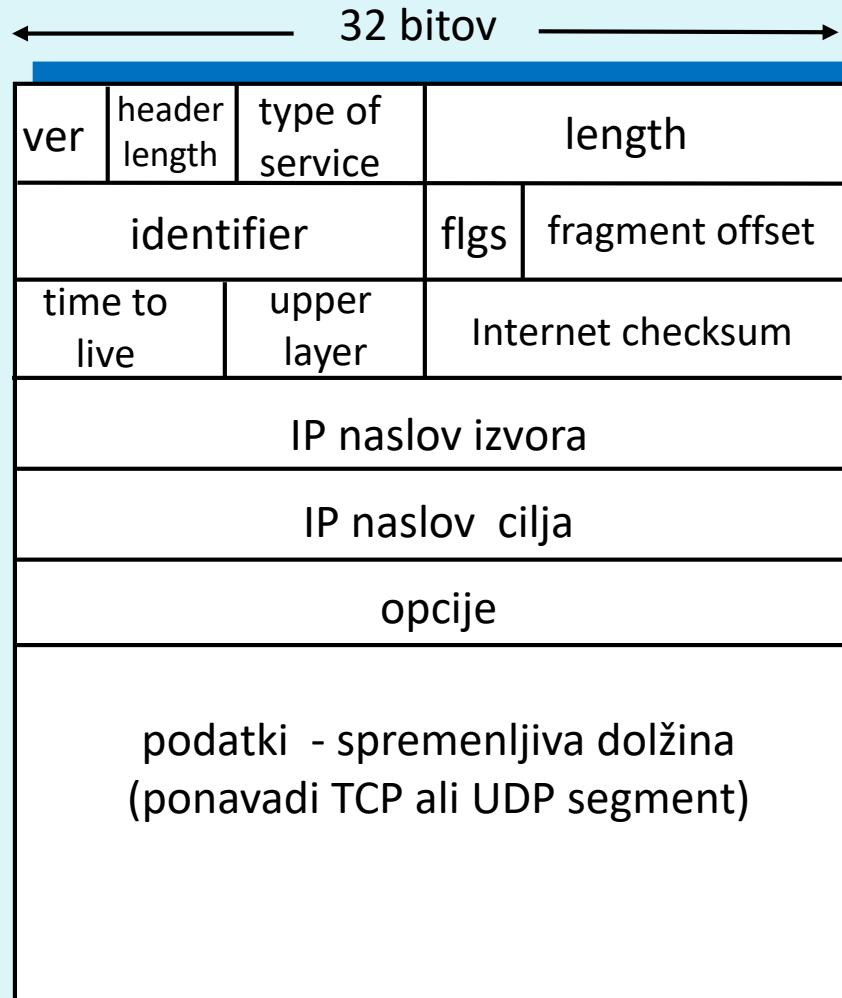
Internet Protocol (IP)



Funkcije omrežne plasti

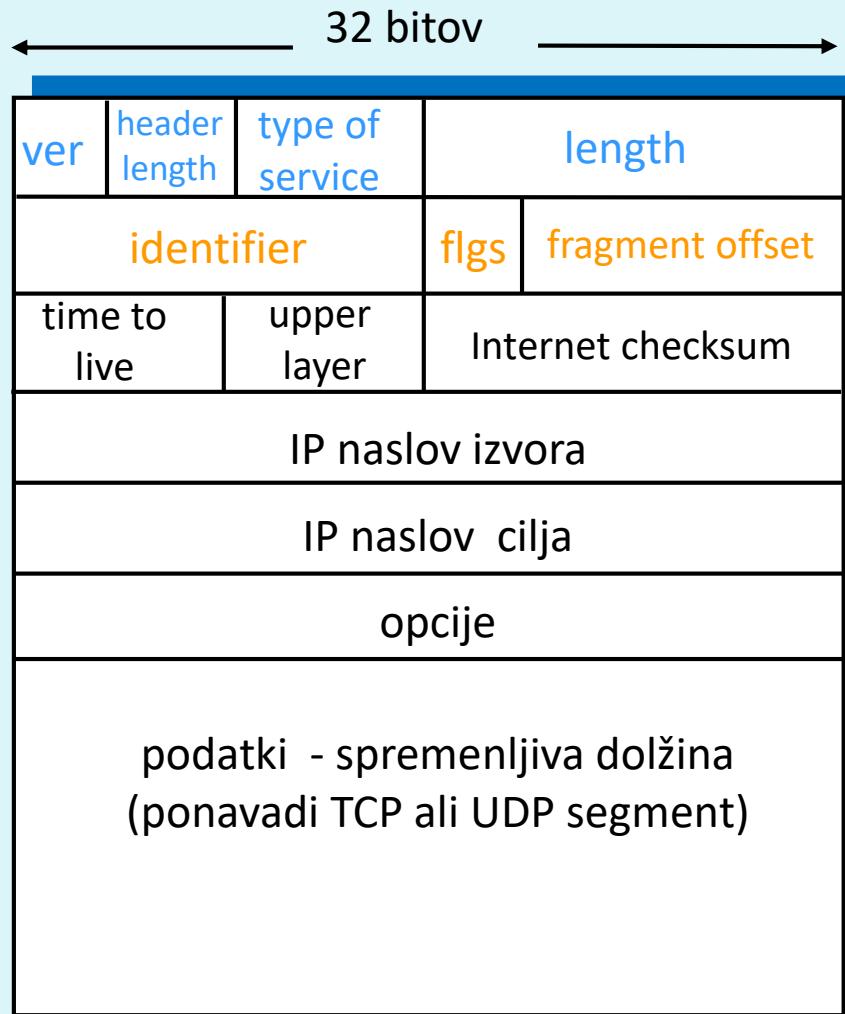


IP protokol: format paketa IPv4



- **VER (4b)**: verzija IP protokola
- **HEADER LENGTH (4b)**: dolžina glave (v 32-bitnih besedah), poda, kje se začnejo podatki
- **TYPE OF SERVICE (8b)**: za razlikovanje datagramov, ki potrebujejo "posebno" obravnavo
- **LENGTH (16b)**: skupna dolžina celega datagrama v Byte-ih (običajno dolžina 1500B)
- **ID, FLAGS, OFFSET (32b)**: potrebno za IP fragmentacijo (razbitje podatkov na več delov)

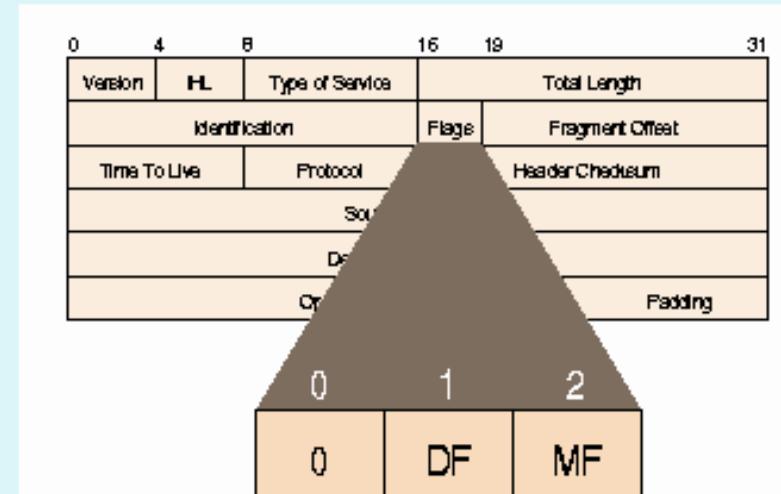
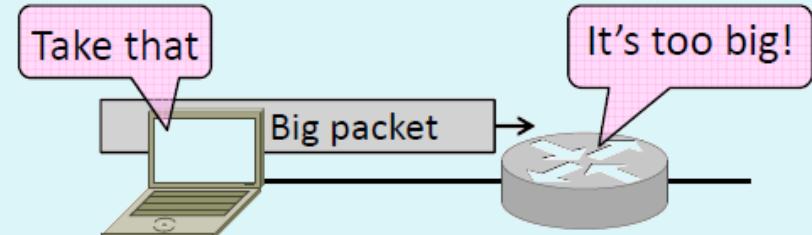
IP protokol: format paketa IPv4



- **TTL (8b):** za preprečitev ciklanja datagramov po omrežju, vsak usmerjevalnik zmanjša vrednost za 1
- **UPPER LAYER (PROTOCOL) (8b):** številka enkapsuliranega protokola v podatkih (6-TCP, 17-UDP)
- **CHECKSUM (16b):** kontrolna vsota (samo) glave datagrama, preračuna jo vsak usmerjevalnik
- **IP naslovi (32b):** naslovi izvora in cilja (začetnega in končnega sistema)
- **OPCIJE (32b):** za možne razširitve glave datagrama (slabosti: večji čas procesiranja, neznana lokacija začetka podatkov; običajno jih ni, glava dolga 20B)
- **PODATKI (spremenljiva dolžina)**

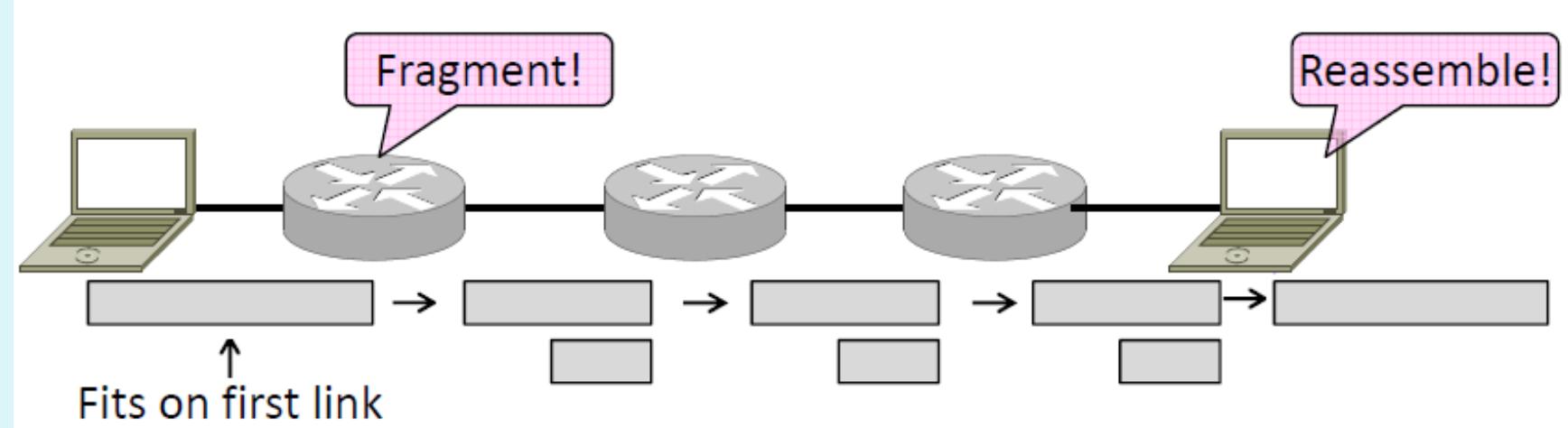
Fragmentacija

- **ZAKAJ?** IP datagrami se morajo enkapsulirati v okvirje povezavne plasti, ti pa imajo omejeno dolžino (MTU, maximum transmission unit, Ethernet: do 1500B, 802.11: 7981B)
- **FRAGMENTACIJA:** (pre)velik paket IP se razbije na več manjših
- **KAKO?** Polja ID, FLAGS, OFFSET se uporabljajo za fragmentacijo
 - bita v zastavicah (FLAGS):
 - DF: don't fragment
 - MF: more fragments



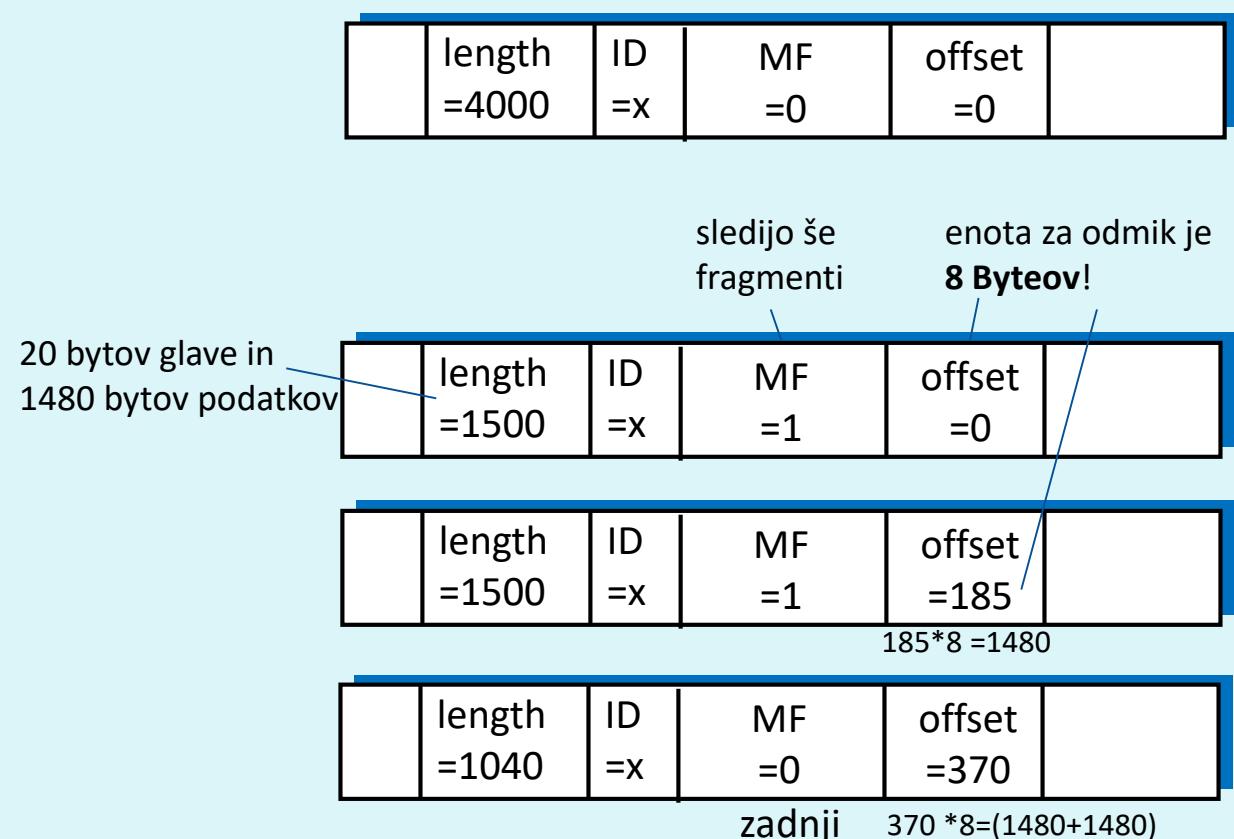
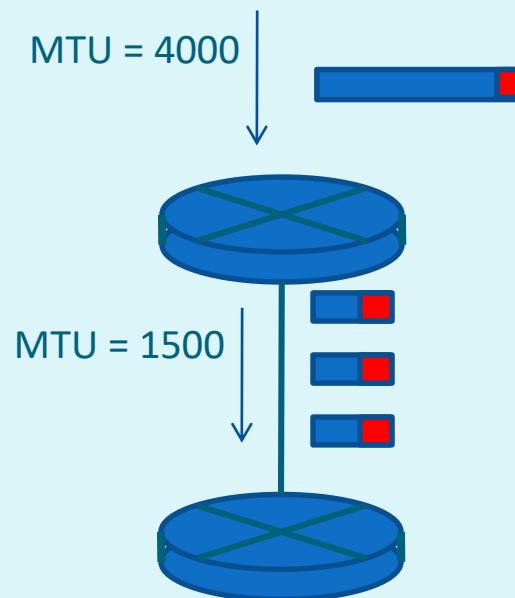
Fragmentacija

- Opombe:
 - v omrežju je lahko več tehnologij, zato se lahko MTU med potjo spreminja. Fragmentacijo lahko izvede tudi usmerjevalnik sredi poti.
 - fragmente združi šele omrežna plast prejemnika pred predajo transportni plasti



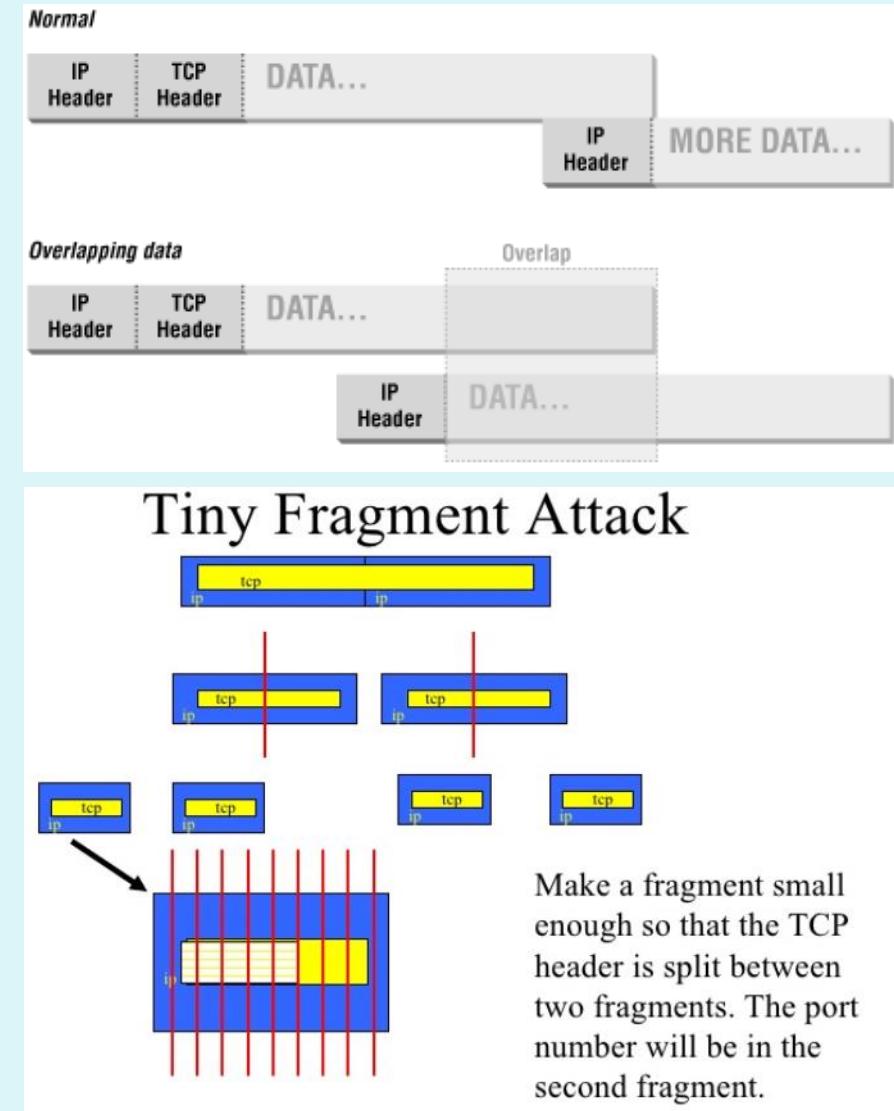
Fragmentacija: primer

- imamo paket dolžine 4000B (torej 20 glava + **3980 podatki**), MTU pa je 1500B
- pri MTU 1500B pomeni, da je 20B glave in **1480B podatkov**
- velikosti podatkov v fragmentih bodo: **1480B + 1480B + 1020B = 3980B**



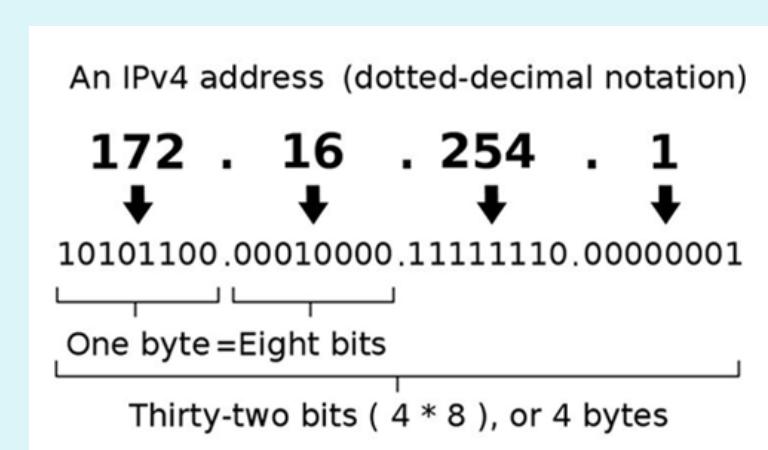
Napadi z uporabo fragmentacije

- obstajajo različni napadi, ki fragmentacijo izkoriščajo, da onemogočijo delovanje omrežnih sistemov (DoS)
 - *overlapping fragment attack*: napadalec fragmentira pakete z namerno napačnimi odmiki/dolžinami (prekrivanje). Pri sestavljanju se ciljni sistem lahko zmede in sesuje (napaka v kodi TCP/IP sklada)
 - *tiny fragment attack*: s fragmentacijo napadalec podatke razkosa tako, da fragmentira tudi podatke v glavi enkapsuliranega protokola. Na ta način ni možno izvesti varnostnega filtriranja po podatkih v glavi.



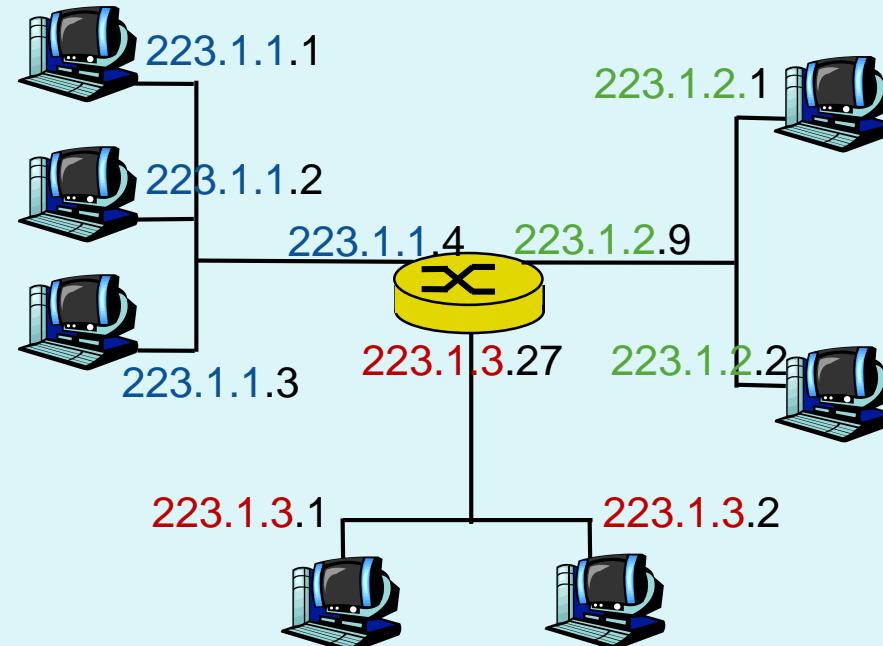
IPv4 nasavljanje

- vmesniki imajo IP (v4) naslove, ki so dolžina 32 bitov
- računalniki imajo običajno en vmesnik, usmerjevalniki več
- primer IPv4 naslova: 11011111 00000001 00000001 00000001
desetiški zapis: 223.1.1.1
- ker je naslov 32-biten, je obstaja cca. 4 milijarde IP naslovov
- naslovi naprav na Internetu morajo biti *globalno* unikatni



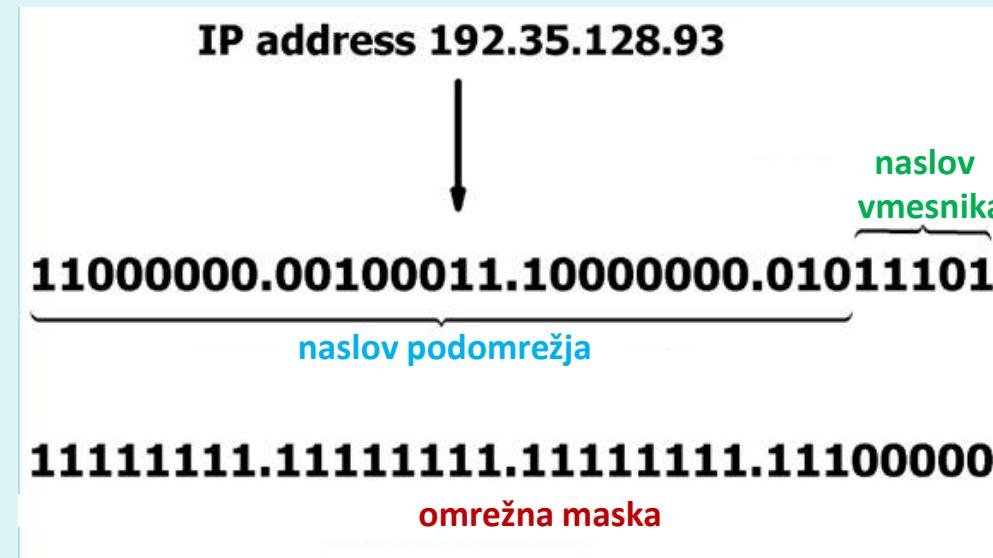
Podomrežje

- IP naslove organiziramo v podomrežja (*subnet*), predpona predstavlja naslov podomrežja
- IP naslovi so smiselno hierarhično organizirani tako, da imajo "lokacijsko sorodne" naprave "podobne" IP naslove
 - analogija: naslovi hiš v isti ulici so podobni



Podomrežje

- **2 dela IP naslova:** naslov omrežja | naslov naprave
- **(Pod)omrežje** je množica vmesnikov, ki imajo enak naslov omrežja, ti vmesniki so medseboj dosegljivi brez posredovanja usmerjevalnika
- **Maska podomrežja** določa dolžino naslova podomrežja (je 32-bitni niz, ki ima enice na mestih, ki označujejo naslov omrežja, na ostalih pa ničle). Okrajšamo jo lahko s številom najbolj pomembnih bitov, npr.: "/27". Primer maske: 11111111 11111111 11111111 11100000 ali 255.255.255.224



Naslavljjanje IPv4

- v Internetu uporabljamo 32-bitne naslove pošiljateljev in prejemnikov (naslov IPv4)
- primer naslova IPv4, binarni in desetiški zapis:

00010010	00011111	10010110	00111110			
18	.	31	.	150	.	62

- naslovi so razdeljeni v predpono (naslov omrežja) in naslov vmesnika
- primer: 20-bitna predpona

naslov:

00010010	00011111	10010110	00111110
----------	----------	----------	----------

omrežna maska:

11111111	11111111	11110000	00000000			
255	.	255	.	240	.	0

naslov omrežja:

00010010	00011111	10010000	00000000			
18	.	31	.	144	.	0

zapis naslova:

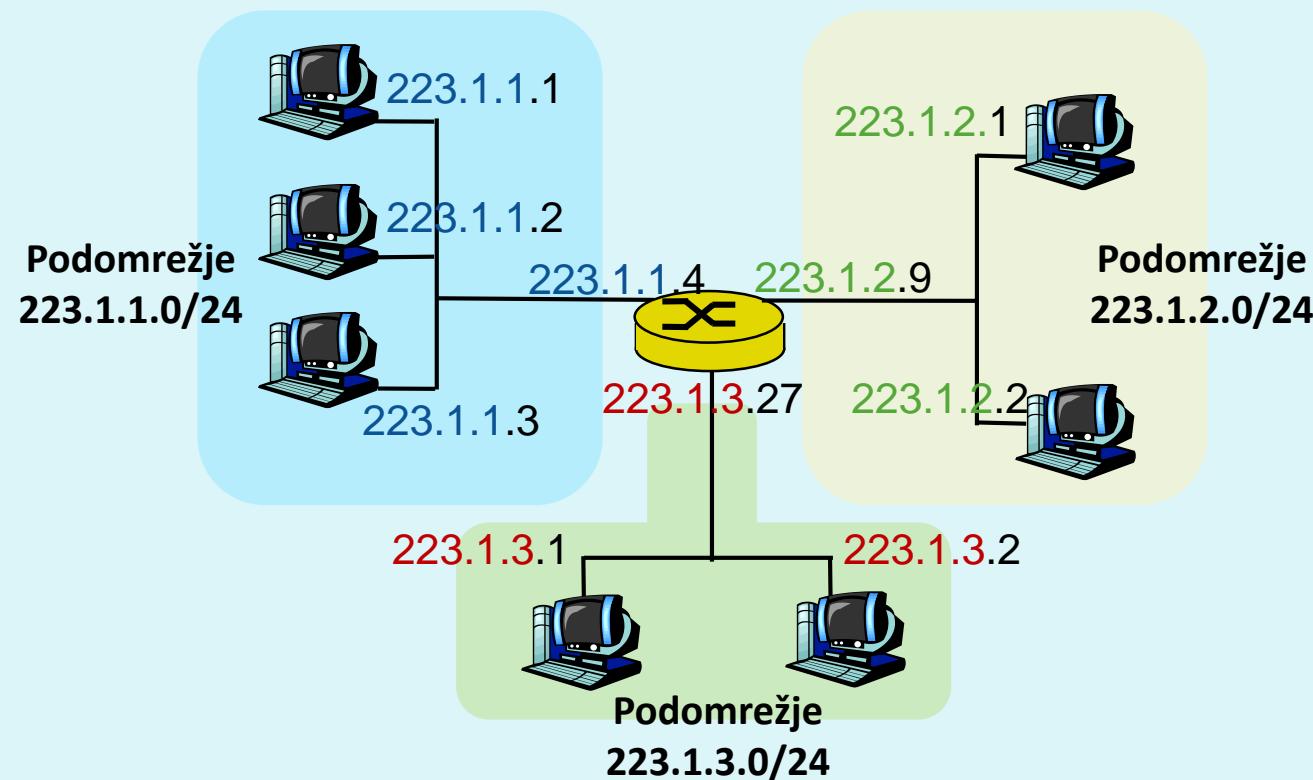
18.31.150.62 / 20

ali:

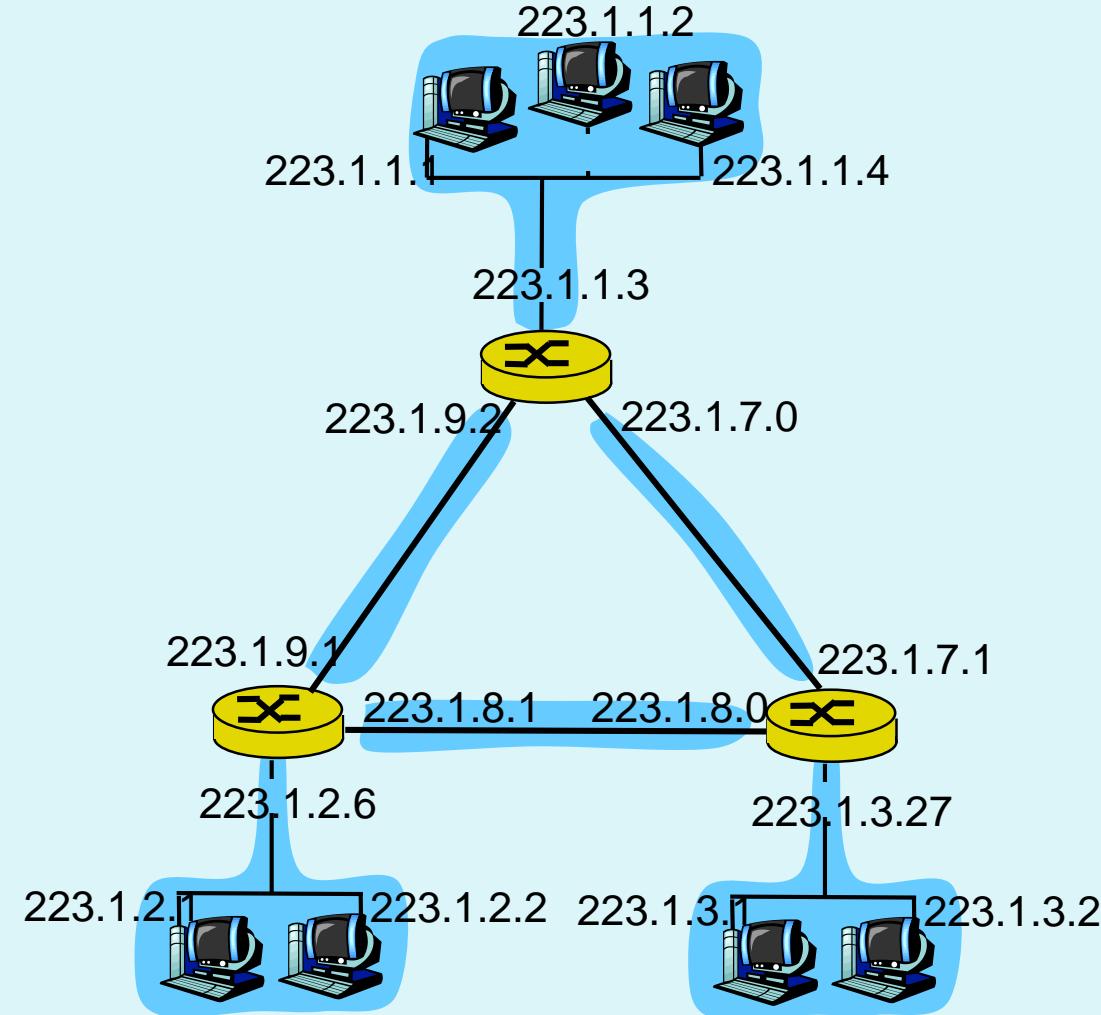
naslov 18.31.150.62, omrežna maska 255.255.240.0

Podomrežje: primer naslavljanja

- usmerjevalnik ima na vsakem vmesniku **drugo podomrežje**
- znotraj (pod)omrežja ni usmerjevalnikov, so pa lahko stikala (switch) in zvezdišča/razdelilniki (hub)

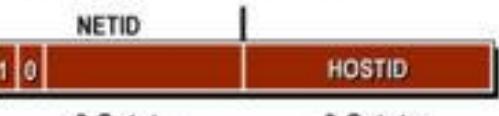


Primer: koliko podomrežij imamo?



Delitev na podomrežja

- sprva definirajo razrede (class) omrežij, ki uporabljajo maske iz 8, 16 ali 24 bitov:

Class	First Octet Range	Max Hosts	Format
A	1-126	16M	
B	128-191	64K	
C	192-223	254	

nesmotrna raba (premalo ali preveč)
naslovov, neuporabljeni iz razreda B
ostanejo globalno neizkoriščeni

- kasneje se vpelje **prefiksna ali CIDR notacija** (Classless Inter-Domain Routing), ki omogoča dodelitev poljubnega števila bitov maski
- "poseben" IP naslov je *broadcast* naslov, ki naslovi vse naprave na podomrežju (naslov naprave je sestavljen iz samih enic, npr. 233.1.1.255 ali 255.255.255.255)

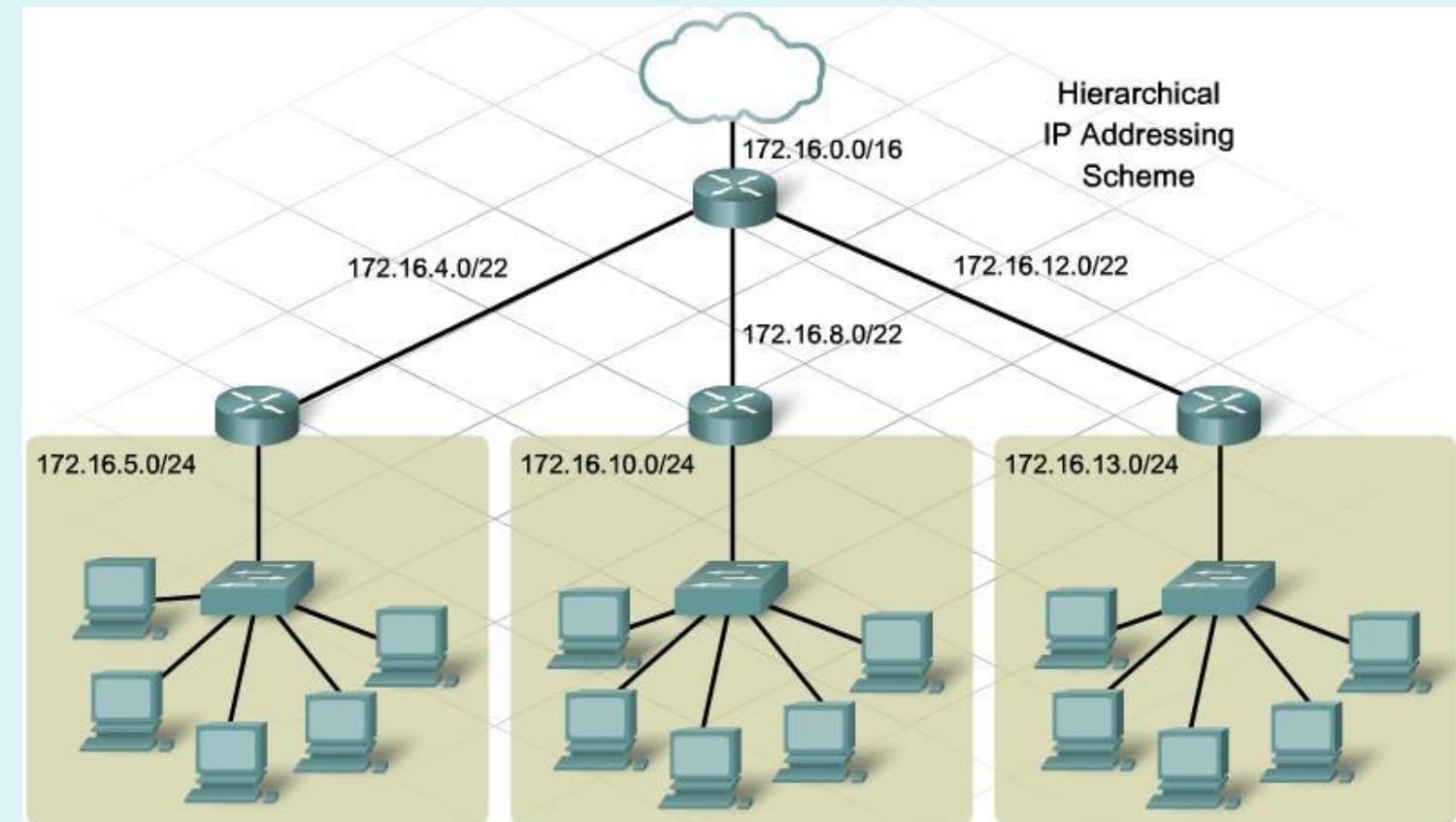
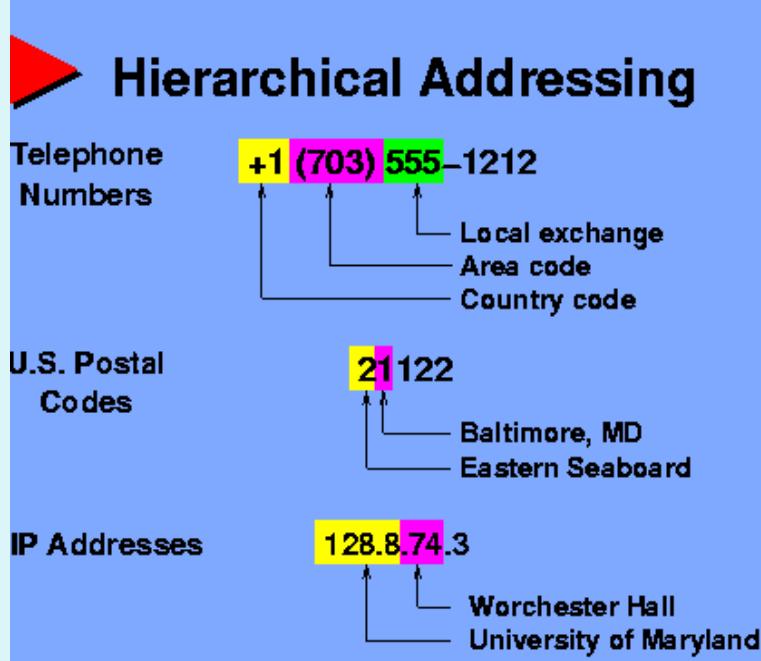
Kako določimo IP naslove?

- **Naprava:**
 - administrator vpiše naslov (fiksen) ali
 - DHCP (*Dynamic Host Configuration Protocol*) strežnik dodeli naslov (dinamičen)
- **Omrežje podjetja:**
 - Ponudnik dostopa do interneta (ISP) dodeli del svojega naslovnega prostora.

ISP-jev blok:	11001000 00010111 00010000 00000000	200.23.16.0/20
Podjetje1:	11001000 00010111 0001 <u>000</u> 0 00000000	200.23.16.0/23
Podjetje2:	11001000 00010111 0001 <u>001</u> 0 00000000	200.23.18.0/23
...

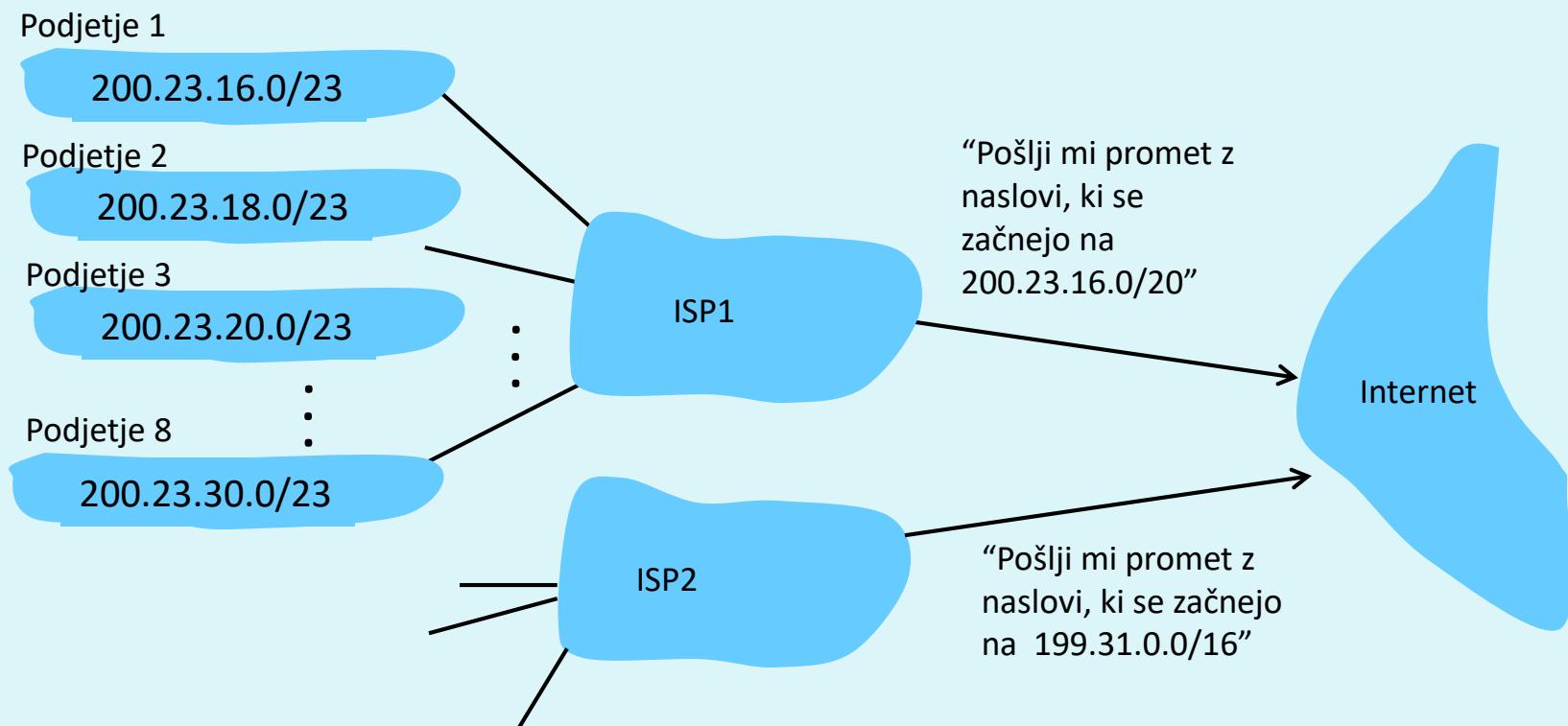
- **Ponudnik interneta (ISP):**
 - dodeli mu ga ICANN (Internet Corporation for Assigned Names and Numbers), neprofitna namenska organizacija, ki vzdržuje tudi korenske DNS strežnike

Hierarhična organizacija naslovov IP



Primer hierarhičnega naslavljanja

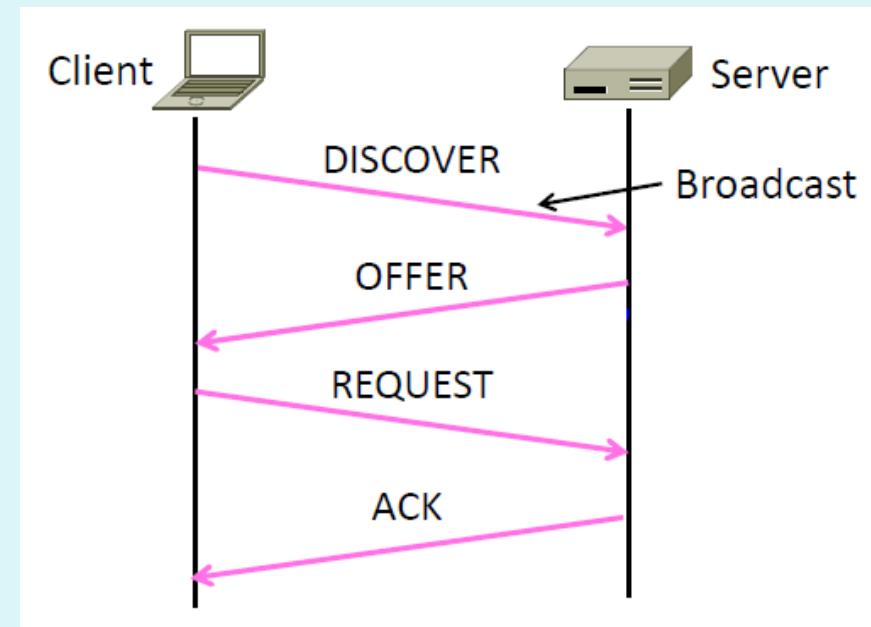
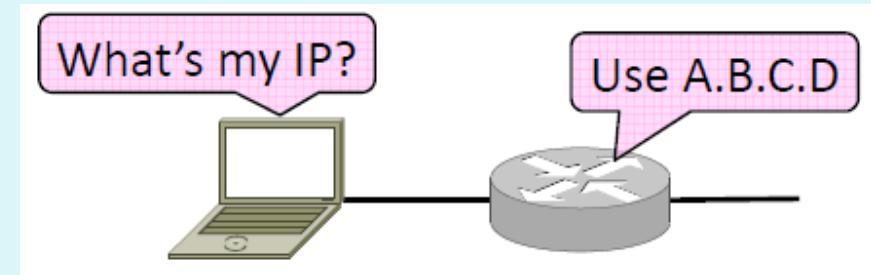
- pravilno dodeljevanje CIDR naslovov olajša usmerjanje: z eno omrežno predpono lahko usmerjamo v več omrežij naenkrat



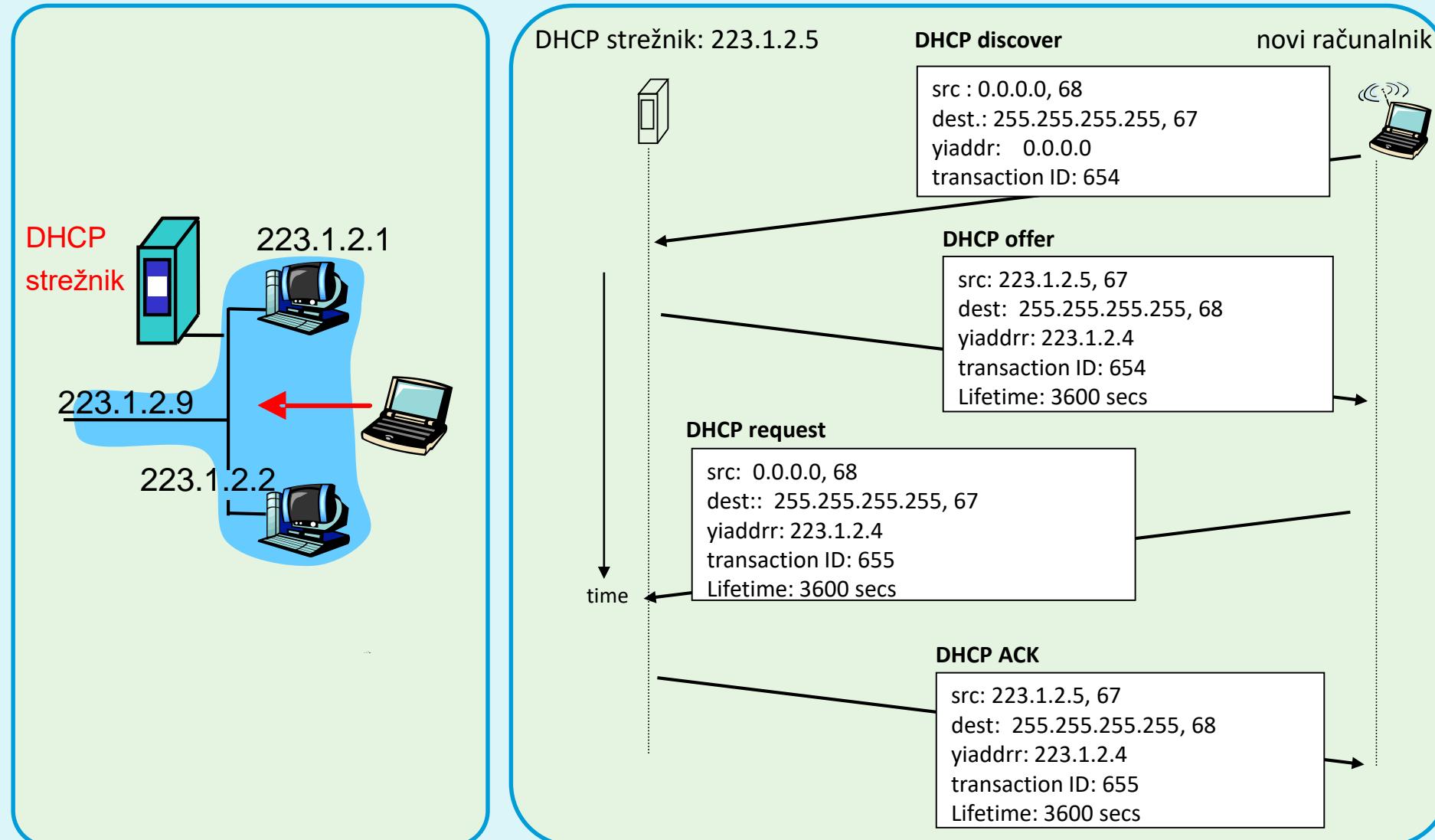
200.23.16.0 = 11001000 00010111 00010000 00000000

Dinamično dodeljevanje: DHCP

- ob priklopu naprava nima naslova IP, potrebna dodelitev osnovnih omrežnih nastavitev
- protokol DHCP (Dynamic Host Configuration Protocol)
- DHCP strežnik dodeli naslov v 4 fazah:
 - DISCOVER
 - OFFER
 - REQUEST
 - ACK



Dinamično dodeljevanje: DHCP strežnik



NAT (*Network Address Translation*)

- NAT: preslikovanje IP naslovov (pre-naslavljanje) se uvede zaradi pomanjkanja IPv4 naslovnega prostora
- namesto, da "trošimo" unikatne javne (**globalne**) naslove, uporabljajmo raje **lokalne naslove**, ki so lahko ponovljivi med različnimi podjetji (ne nastopajo v javnem internetu)
- zasebni naslovni prostor

Naslovi	Omrežje/maska	Št. naslovov
10.0.0.0 - 10.255.255.255	10.0.0.0/8	2^{24}
172.16.0.0 - 172.31.255.255	172.16.0.0/12	2^{20}
192.168.0.0 - 192.168.255.255	192.168.0.0/16	2^{16}

- usmerjevalnik uporabi NAT, da lokalni naslov preslika v globalni

NAT: primer delovanja

2: NAT usmerjevalnik spremeni naslov izvora 10.0.0.1, 3345 v 138.76.29.7, 5001, In to vstavi v tabelo

NAT preslikovalna tabela: IP, port	
Naslovi WAN strani	Zasebni naslovi
138.76.29.7, 5001	10.0.0.1, 3345
.....

1: rač. 10.0.0.1 pošlje paket na 128.119.40, 80

2: NAT usmerjevalnik spremeni naslov izvora 10.0.0.1, 3345 v 138.76.29.7, 5001, In to vstavi v tabelo

3: Odgovor s ciljnim naslovom 138.76.29.7, 5001

S: 10.0.0.1, 3345
D: 128.119.40.186, 80

10.0.0.4

S: 128.119.40.186, 80
D: 10.0.0.1, 3345

4: NAT usmerjevalnik spremeni naslov cilja v 10.0.0.1, 3345

10.0.0.1

10.0.0.2

10.0.0.3

S: 128.119.40.186, 80
D: 138.76.29.7, 5001

138.76.29.7

3

1

4

Prednosti in slabosti pristopa NAT

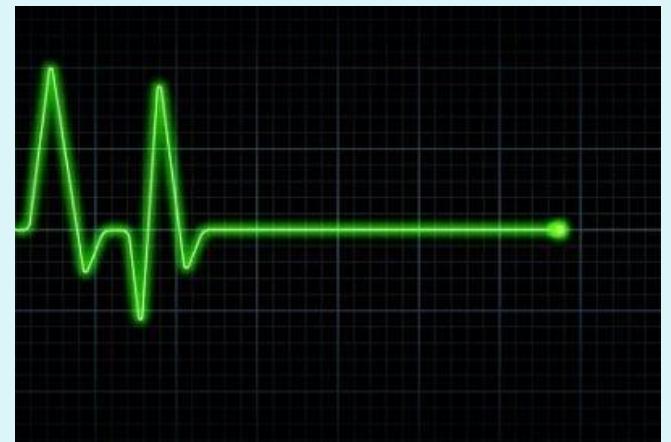
PREDNOSTI

- zadošča samo 1 javni naslov za dostop celega omrežja do Interneta
- naslove notranjih naprav in ponudnika interneta (!) lahko **spreminjamamo neodvisno** od zunanjega naslova
- večja **varnost** notranjih naprav, ker niso javno dostopne
- 16-bitno polje za vrata (port) omogoča evidentiranje cca. 60.000 povezav do notranjih naprav

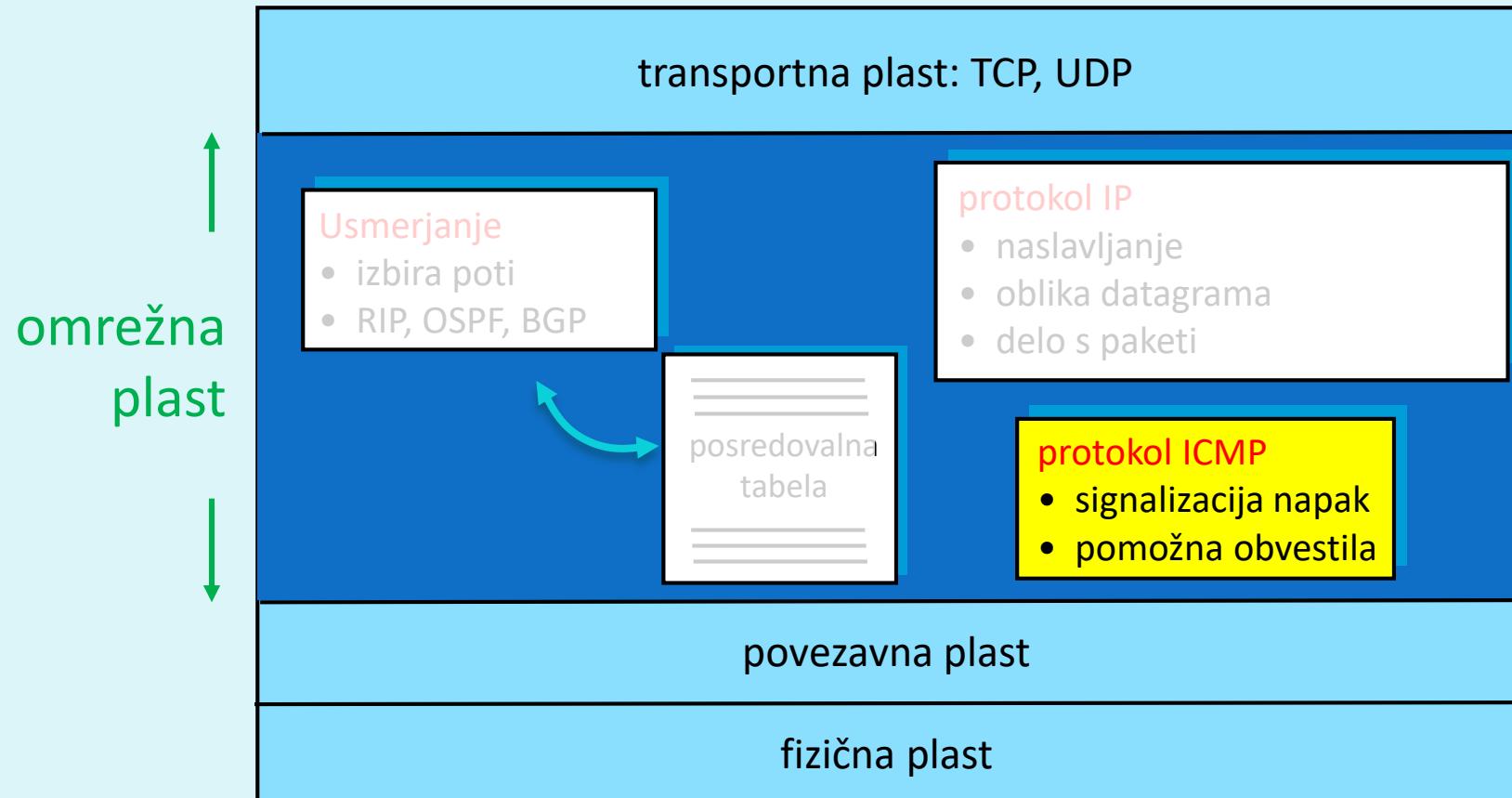
KRITIKA

- usmerjevalniki **naj bi delali na 3. plasti** (torej ne bi imeli opravka z vrati - ki so del 4. plasti!)
- vrata (porti) so namenjeni **naslavljjanju procesov**, ne računalnikov
- krši **princip končnih sistemov** (*end-to-end argument*), ki zahteva, da je za aplikacije omrežje transparentno; težavo imamo pri P2P aplikacijah, do katerih znotraj NATa ni možno dostopiti.
- za reševanje pomanjkanja naslovov je **bolje uporabiti IPv6!**

Internet Control Message Protocol (ICMP)



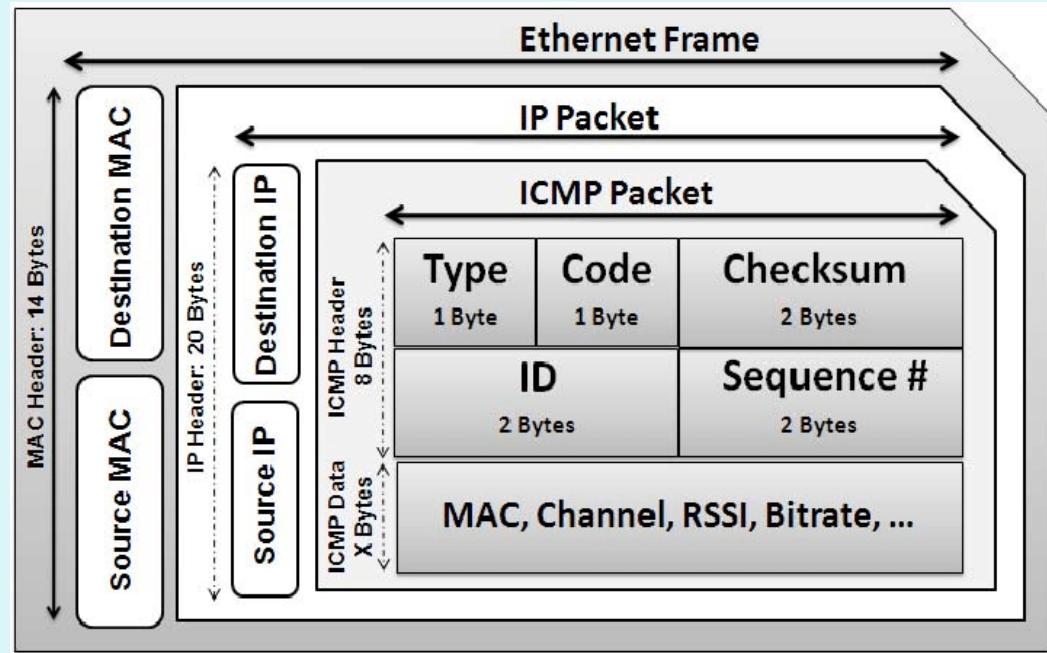
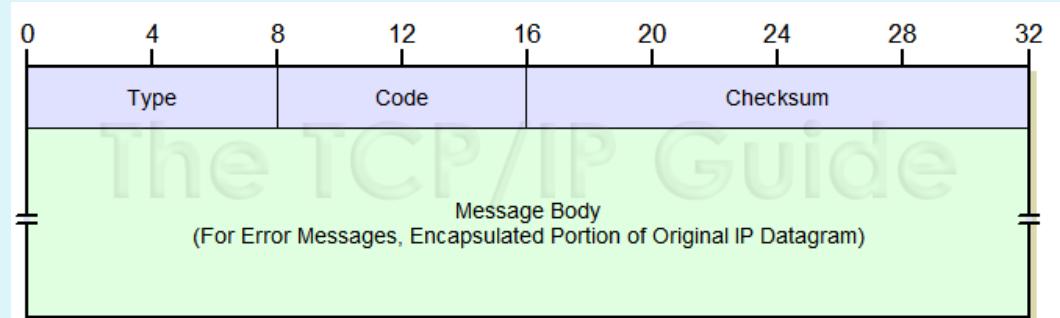
Funkcije omrežne plasti



ICMP

- *Internet Control Message Protocol (ICMP)*
- sporočilo, enkapsulirano **znotraj paketa IP** (enako kot protokoli na transportni plasti)
- uporablja se za izmenjavo sporočila v zvezi z omrežjem: napake, nedosegljivost, protokol, vrata

Tip	Koda	Pomen
0	0	echo reply (ping)
3	0	dest network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header



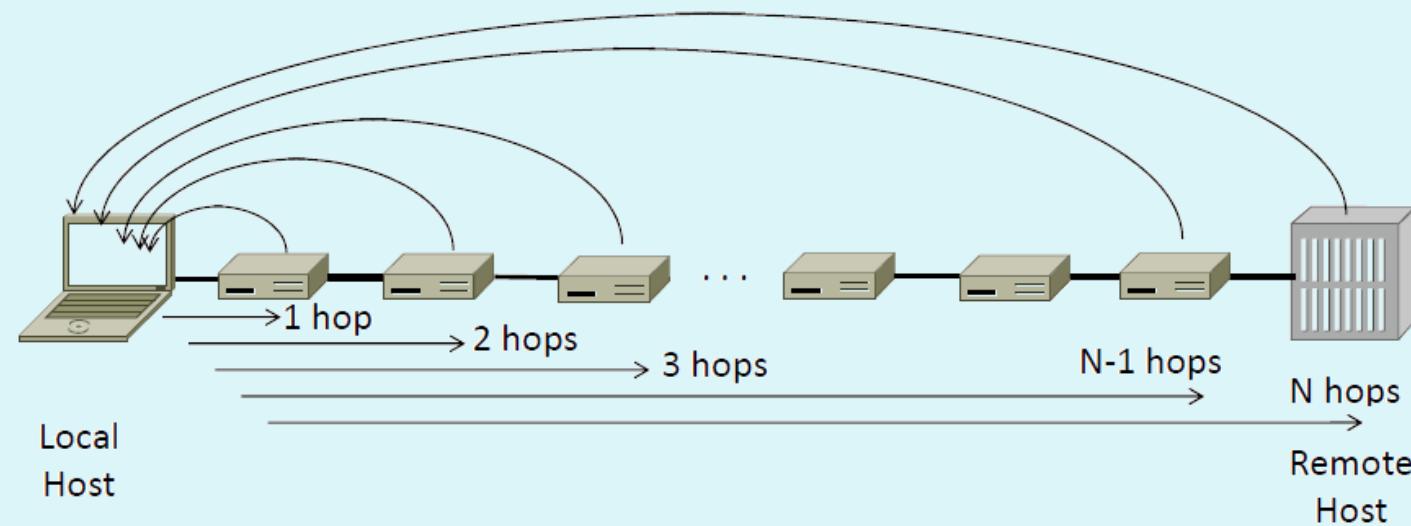
Traceroute: aplikacija, ki uporablja ICMP

- **traceroute** analizira, po kateri poti gre promet do določenega cilja
 - razvil ga je Van Jacobson, leta 1987
 - koračni protokol, preizkuša komunikacijo na posameznih korakih (hop-ih) do cilja, da poišče pot skozi omrežje

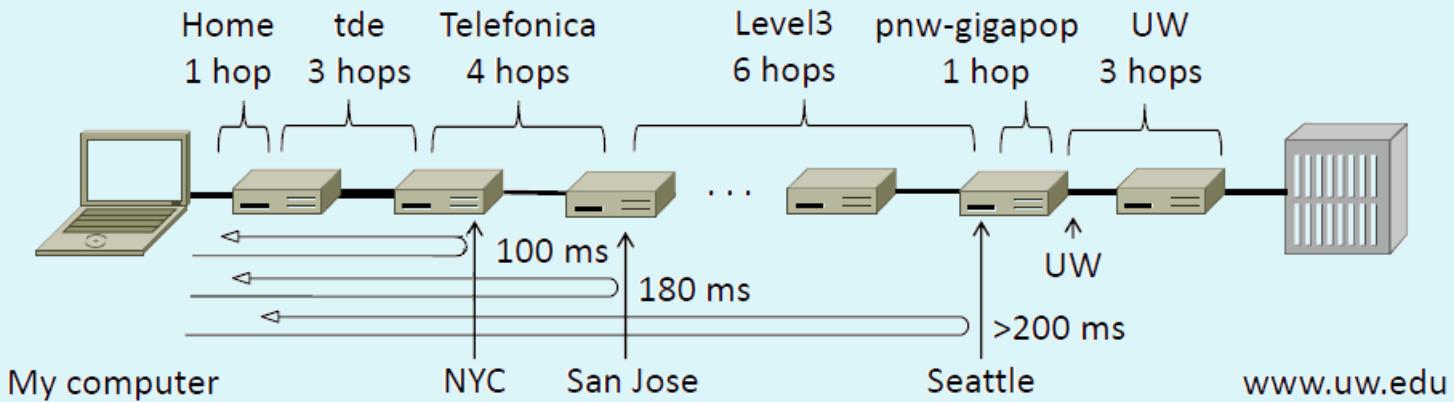


Delovanje aplikacije Traceroute

- izvor pošilja serijo paketov cilju, prvi ima TTL=1, drugi TTL=2 itd.
 - paket vsebuje zahtevek ICMP **echo request** (tip=8, koda=0)
- usmerjevalnik pogleda vrednost TTL:
 - če TTL \neq 0, usmerjevalnik zmanjša vrednost TTL za 1 in posreduje naprej,
 - če TTL=0, usmerjevalnik paket zavrže in pošlje izvoru obvestilo ICMP **TTL expired** (tip=11, koda=0), ki vključuje naslov usmerjevalnika
- za vsako prejeto ICMP sporočilo izvor izračuna tudi čas vrnitve in statistike



Traceroute: primer



```
c:\ Administrator: C:\Windows\system32\cmd.exe
>
> tracert www.stanford.edu
Tracing route to www-v6.stanford.edu [171.67.215.200]
over a maximum of 30 hops:
 1  <1 ms    <1 ms    <1 ms  Broadcom.amis.lan [192.168.1.254]
 2  1 ms     1 ms     1 ms  asr-lj1.amis.net [212.18.32.174]
 3  1 ms     1 ms     1 ms  mx-lj1-te-0-3-0.amis.net [212.18.44.161]
 4  8 ms     8 ms     8 ms  mx-vi1-te-0-0-0.amis.net [212.18.44.142]
 5  23 ms    23 ms    23 ms  30gigabitethernet4-3.core1.fra1.he.net [80.81.192.172]
 6  32 ms    33 ms    40 ms  10gigabitethernet1-4.core1.par2.he.net [184.105.213.162]
 7  122 ms   110 ms   121 ms  10gigabitethernet7-1.core1.ash1.he.net [184.105.213.93]
 8  181 ms   180 ms   187 ms  10gigabitethernet11-1.core1.pao1.he.net [184.105.213.177]
 9  186 ms   181 ms   181 ms  stanford-university.10gigabitethernet1-4.core1.pao1.he.net [216.218.209.1]
118]  187 ms   187 ms   187 ms  boundarya-rtr.Stanford.EDU [68.65.168.33]
11  *        *        *      Request timed out.
12  *        *        *      Request timed out.
13  186 ms   187 ms   186 ms  www.stanford.edu [171.67.215.200]
Trace complete.
>
```

Internet Protocol v6 (IPv6)



Internet Protocol ver. 6 (IPv6) - motivacija

1. potreben je **večji naslovni prostor** (IPv4 naslovi, ki so 32-bitni, so že izčrpani)
 - naslovi dolžine 128-bitov omogočajo $340,282,366,920,938,463,463,374,607,431,768,211,456$ naslovov (3.4×10^{38}) oziroma 6.65×10^{23} naslovov na m² zemljine površine
 - tipičen (unicast) naslov: 64 bitov je ID podomrežja + 64 bitov za ID vmesnika
2. potrebno **hitrejše usmerjanje**
 - fiksna glava 40 Byte-ov, opcij nimamo (in s tem spremenljive dolžine glave!)
 - fragmentacija ni dovoljena, ker upočasnjuje procesiranje
3. potrebno zagotavljanje **kakovosti storitev** (QoS) za posebne tokove podatkov
 - oznaka "vrste toka" (flow label) v paketu IPv6

RIPE NCC Begins to Allocate IPv4 Address Space From the Last /8

14 Sep 2012

On Friday 14 September, 2012, the RIPE NCC, the Regional Internet Registry (RIR) for Europe, the Middle East and parts of Central Asia, distributed the last blocks of IPv4 address space from the available pool.

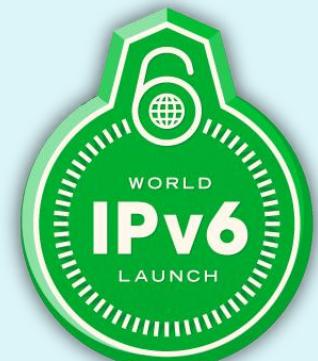
This means that we are now distributing IPv4 address space to Local Internet Registries (LIRs) from the last /8 according to section 5.6 of "IPv4 Address Allocation and Assignment Policies for the RIPE NCC Service Region".

This section states that an LIR may receive one /22 allocation (1,024 IPv4 addresses), even if they can justify a larger allocation. This /22 allocation will only be made to LIRs if they have already received an IPv6 allocation from an upstream LIR or the RIPE NCC. No new IPv4 Provider Independent (PI) space will be assigned.

It is now imperative that all stakeholders deploy IPv6 on their networks to ensure the continuity of their online operations and the future growth of the Internet.

More information on IPv6 and its deployment, advice from experts and where to get training

Dostopno na: <http://www.ripe.net/internet-coordination/news/announcements/ripe-ncc-begins-to-allocate-ipv4-address-space-from-the-last-8>



Sintaksa IPv6 naslova

- IPv6 naslov v binarni obliki:

0010000111011010000000001101001100000000000000001011100111011
0000001010101010000000001111111111110001010001001110001011010

- razdeljen na osem 16-bitnih skupin:

0010000111011010 0000000011010011 0000000000000000 001011100111011
0000001010101010 0000000011111111 111111000101000 1001110001011010

- zapisan šestnajstško, ločeno z dvopičji

`21da:00d3:0000:2f3b:02aa:00ff:fe28:9c5a`
ali `21da:d3:0:2f3b:2aa:ff:fe28:9c5a` (vodilne 0 izpustimo)

- zaporedje (celih!) 16-bitnih blokov iz samih ničel lahko zapišemo kot dve dvopičji ::

- `fe80:0:0:0:2aa:ff:fe9a:4ca2` ali krajše `fe80::2aa:ff:fe9a:4ca2`
- `ff02:0:0:0:0:0:0:2` ali krajše `ff02::2`
- `ff02:30:0:0:0:0:0:5` ni isto kot `ff02:3::5` (lahko pa zapišemo `ff02:30::5`)

- kompatibilnost z v4 naslovi: spredaj dodamo ničle

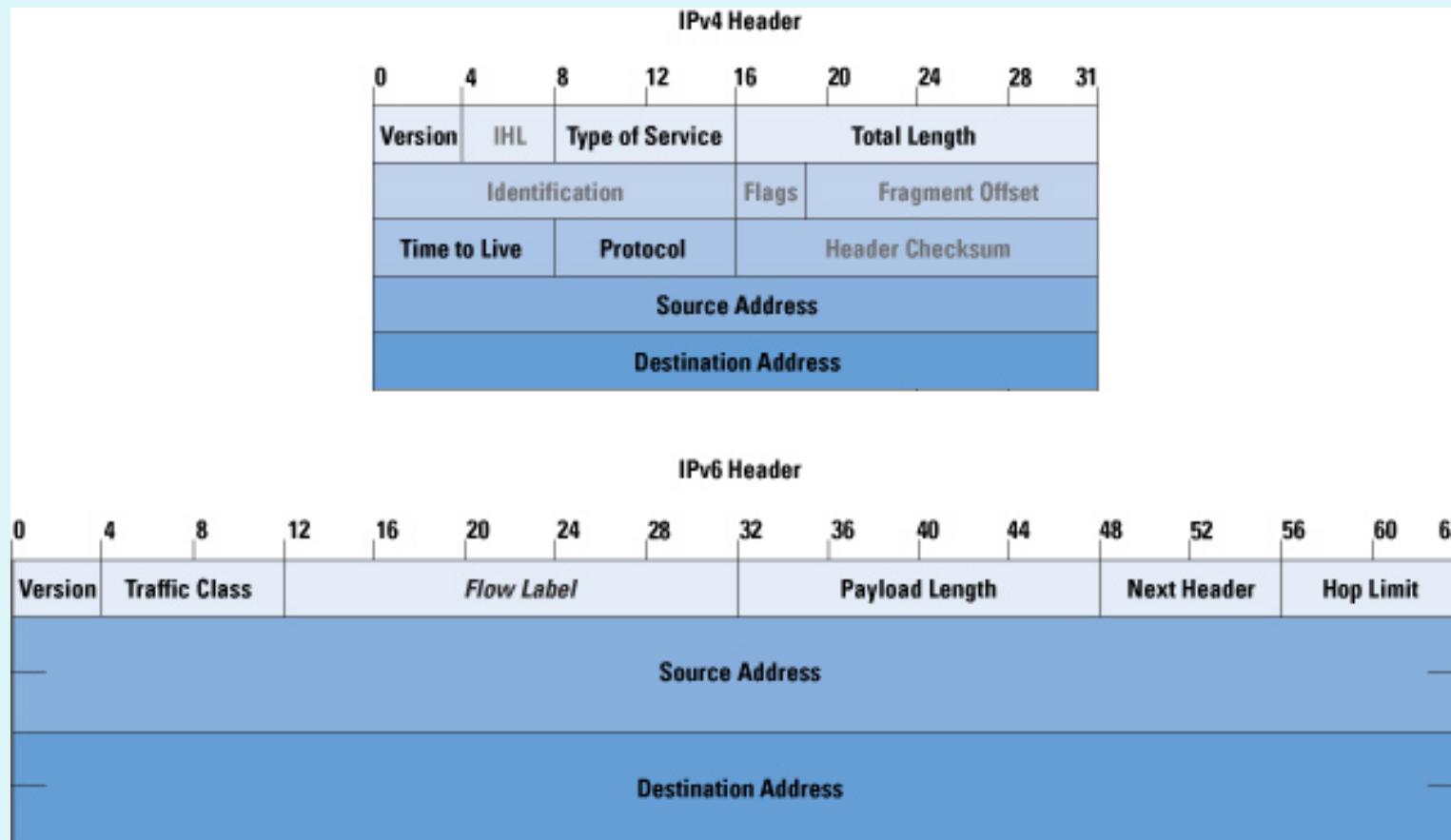
- `193.2.72.1` → `::193.2.72.1` (lahko celo pustimo pike iz IPv4 naslova!)

IPv6: hitrejše procesiranje paketov

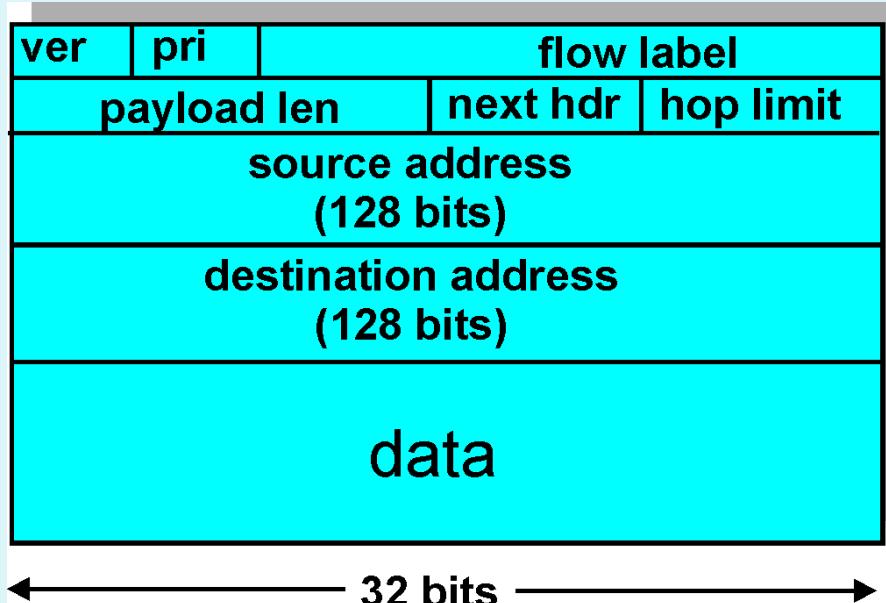
- **fragmentacija se ne izvaja več** (za delitev paketov skrbita pošiljatelj in sprejemnik). V primeru, da je paket prevelik, usmerjevalnik zavrže paket in sporoči pošiljatelju "Packet Too Big",
 - nova verzija protokola ICMP: **ICMPv6** (eno od sporočil tega protokola je zgornji "Packet Too Big"),
- glava več **ne vsebuje kontrolne vsote**, ker je ta prisotna že v enkapsuliranih protokolih znotraj IP paketa in zavira hitro procesiranje (preračunavanje ob spremembi TTL),
- polja za **opcije v glavi ni več**. Možno jih je implementirati kot poseben enkapsuliran protokol (uporaba polja next header).

Primerjava paketov IPv4 in IPv6

- Katera glava je bolj preprosta?



IPv6 format paketa



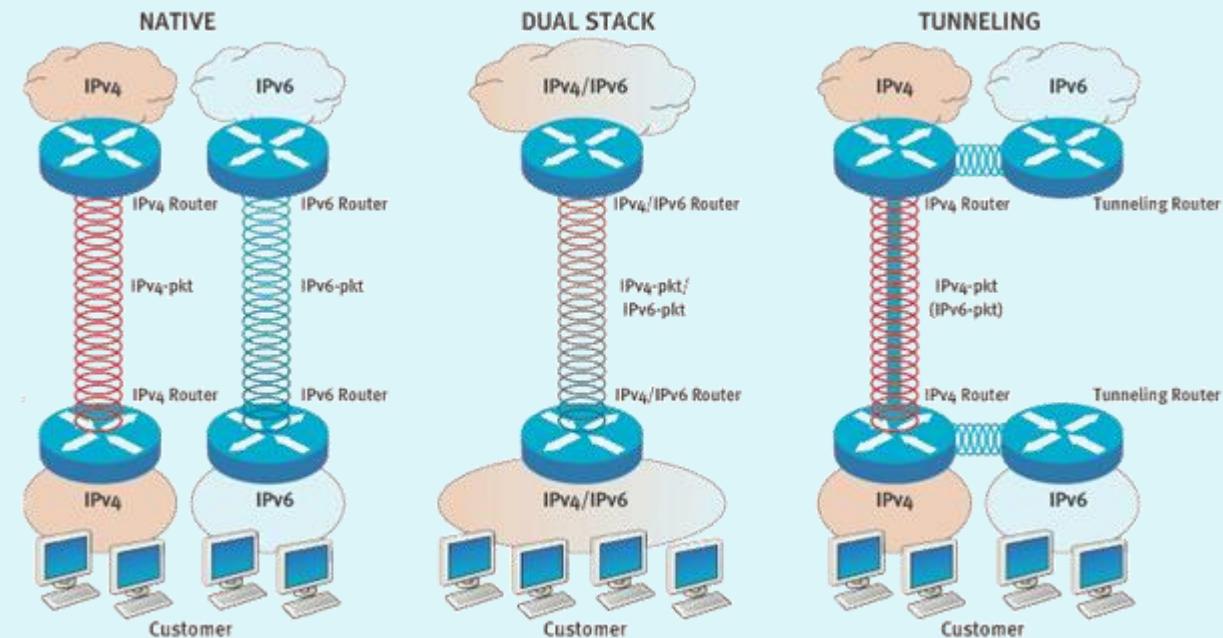
- ni polj za fragmentacijo, kontrolne vsote in opcij
- hitro procesiranje glave paketov



- **VER** (4b): verzija IP protokola (6)
- **PRI** ali **TRAFFIC CLASS** (8b): podobno kot Type Of Service pri IPv4, oznaka prioritete za posebne pakete / aplikacije
- **FLOW LABEL** (20b): oznaka "toka" podatkov, ki omogoči posebno zagotavljanje kakovosti storitve (npr. audio/video)
- **PAYLOAD LENGTH** (16b): velikost podatkov, ki sledijo glavi
- **NEXT HDR** (8b): tip enkapsuliranega protokola
- **HOP LIMIT** (8b): enako kot TTL

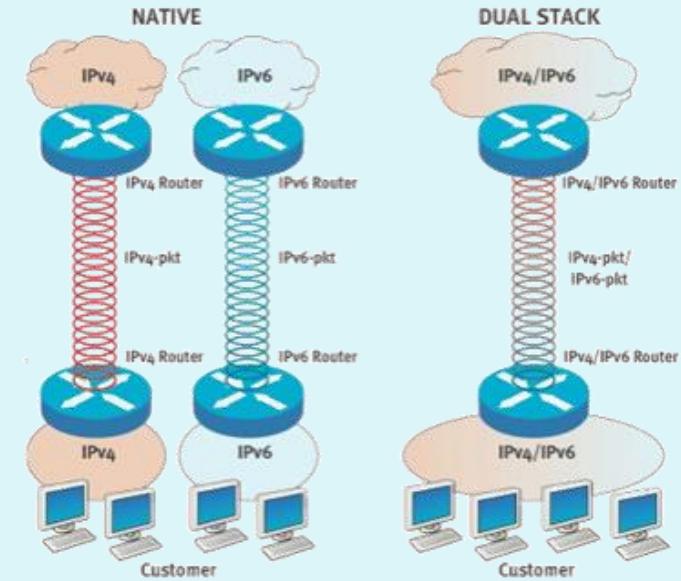
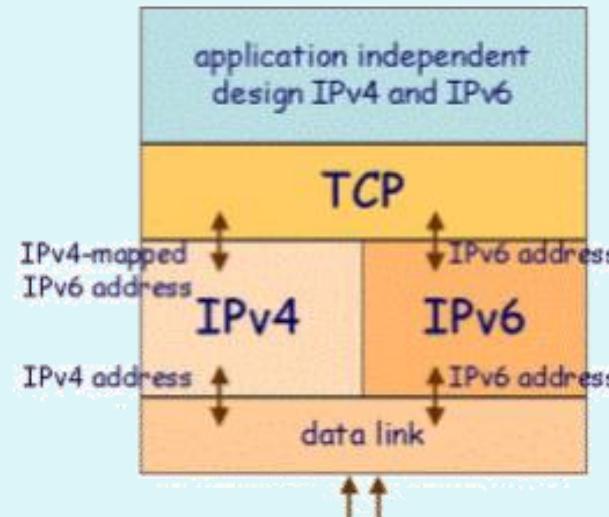
Prehod iz IPv4 -> IPv6

- vseh naprav ni mogoče nadgraditi naenkrat ("flag day")
- za prehod se najpogosteje uporablja dve rešitvi:
 - dvojni sklad (*dual-stack*, vozlišča uporabljajo vzporedni implementaciji IPv4 in IPv6)
 - tuneliranje (*tunneling*, paket IPv6 zapakiramo v paket IPv4 kot podatke)



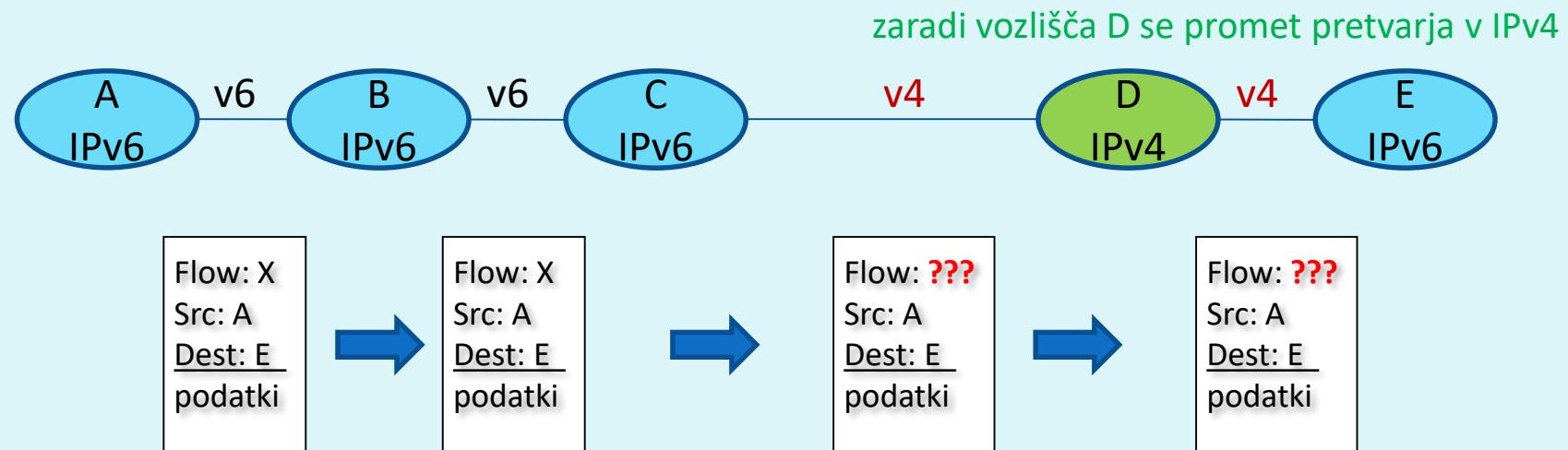
Prehodni mehanizem: Dvojni sklad

- usmerjevalnik "govori" **IPv4 in IPv6**. Z vozlišči, ki uporablja IPv6 komunicira s tem protokolom, sicer z IPv4
- kako naprava ugotovi, ali uporabljati IPv4 ali IPv6?
Pri povpraševanju za IP, **DNS strežnik** (Domain Name Server) vrne IPv6 ali IPv4 naslov.



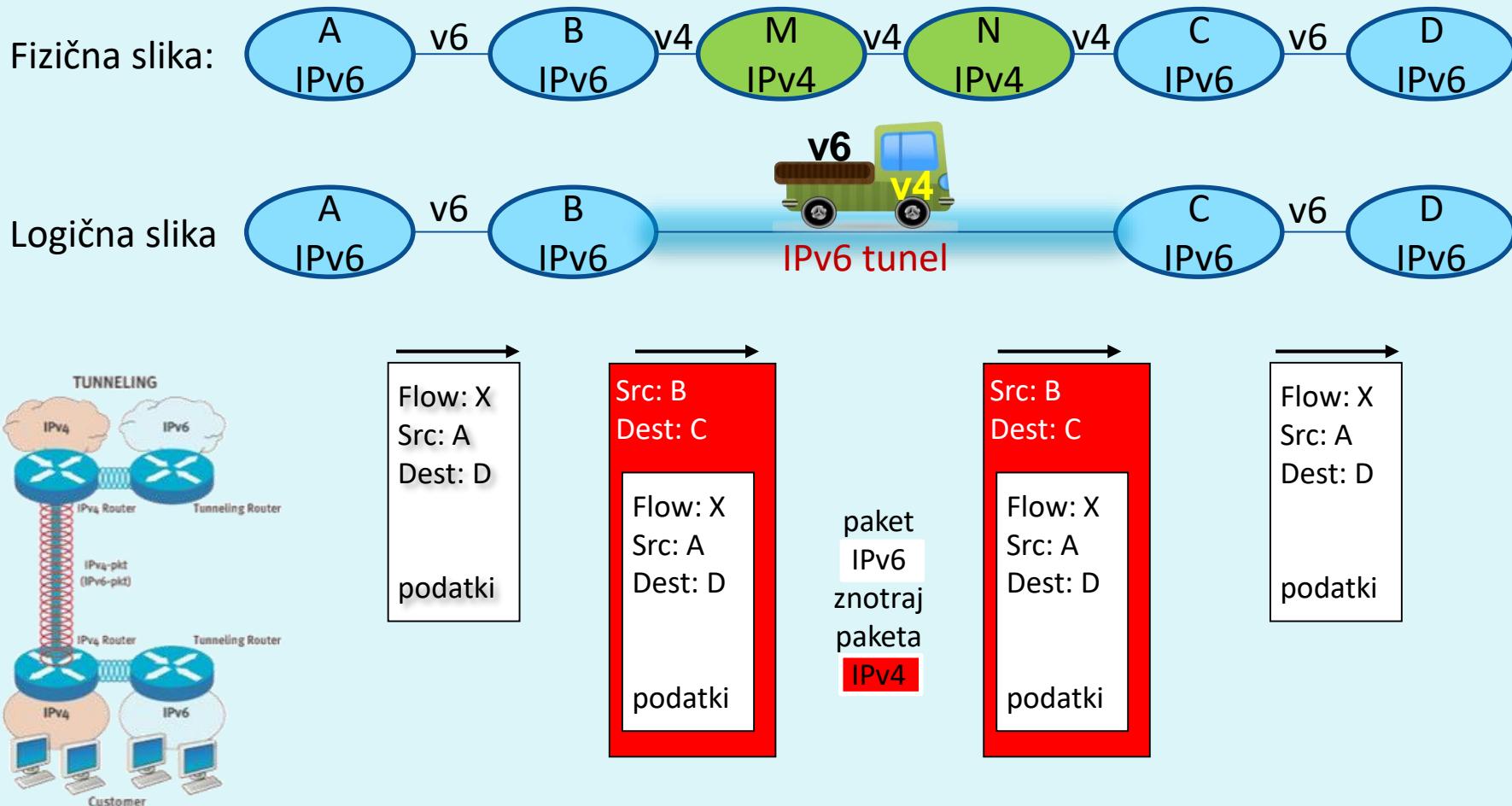
Prehodni mehanizem: Dvojni sklad

- če je na poti med dvema IPv6 vozliščema kakšno IPv4, se bo promet vmes **pretvarjal v IPv4**; Specifična polja protokola IPv6 (flow label?) se bodo izgubila!



IPv4 -> IPv6: Tuneliranje

- paket IPv6 zapakiramo v paket IPv4 kot podatke



Varnost na omrežni plasti: IPSec

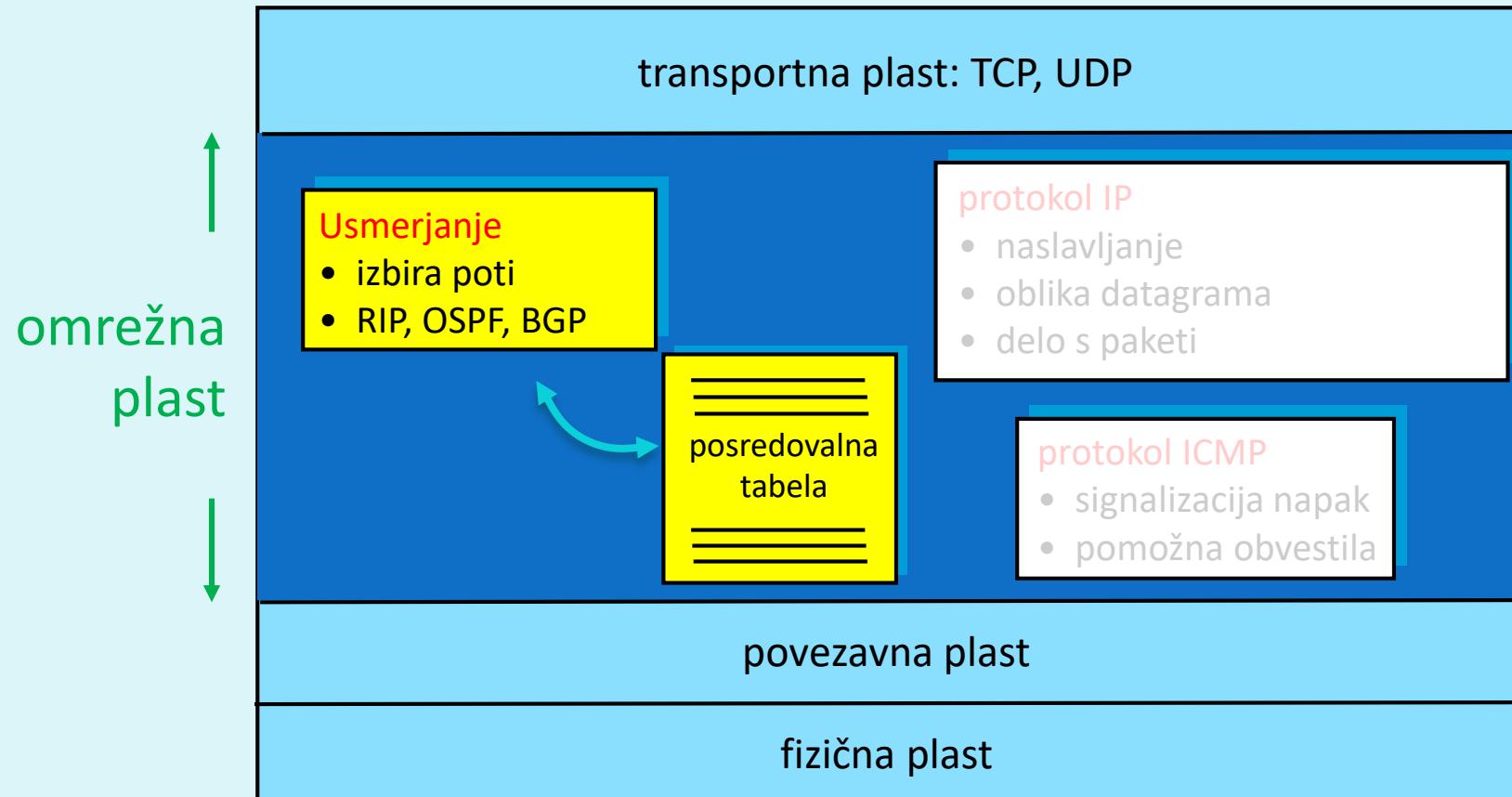
- komunikacija na omrežni plasti poteka nevarovano (možna so ponarejanja izvornih naslovov, prisluškovanje/ponarejanje komunikacije ipd.)
- IPSec: nabor protokolov, ki skrbijo za varno komunikacijo na omrežnem nivoju (AH, ESP)
- storitve:
 - dogovor o uporabljenih kriptografskih algoritmih in ključih
 - enkripcija in dekripcija (zakrivanje komunikacije)
 - integriteta podatkov
 - avtentikacija izvora



Usmerjanje

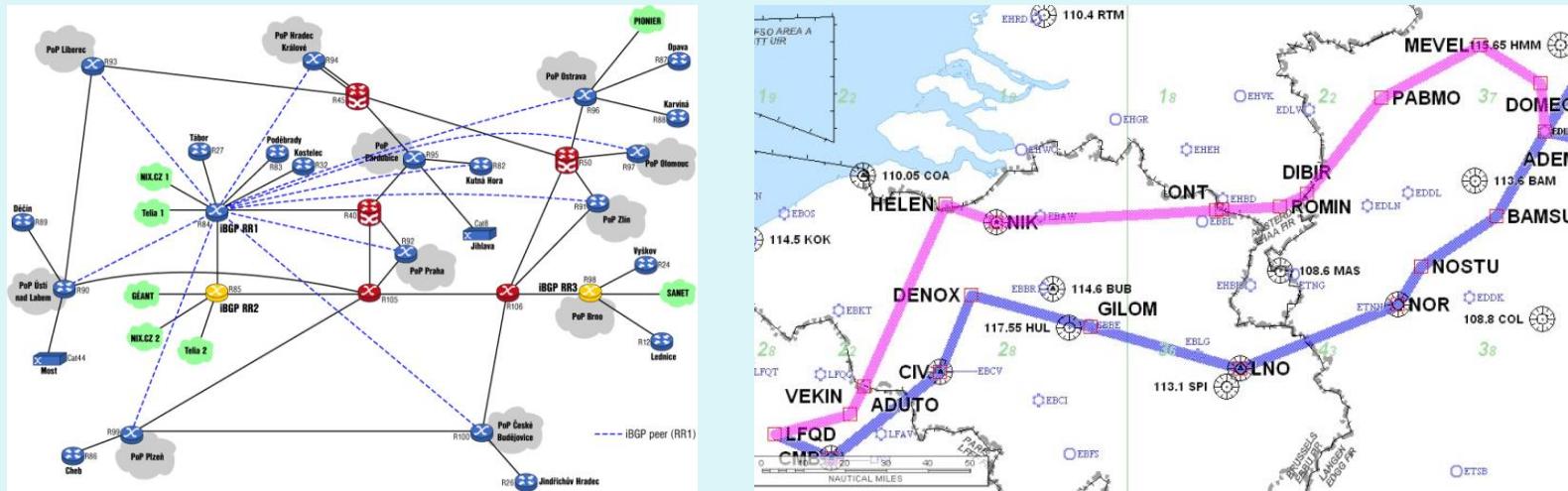


Funkcije omrežne plasti



Usmerjanje

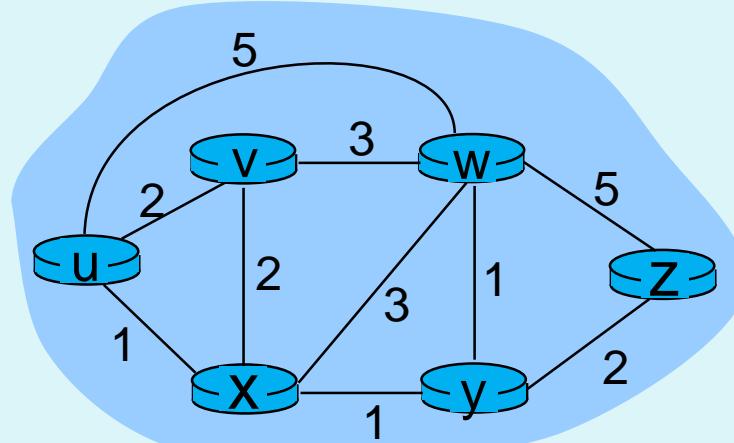
- problem: po kateri poti naj paket potuje od izvora do cilja?



- omrežja modeliramo s teorijo grafov $G = \langle V, P \rangle$:
 - vozlišča V : usmerjevalniki
 - povezave P : komunikacijske povezave

Usmerjevalni protokoli

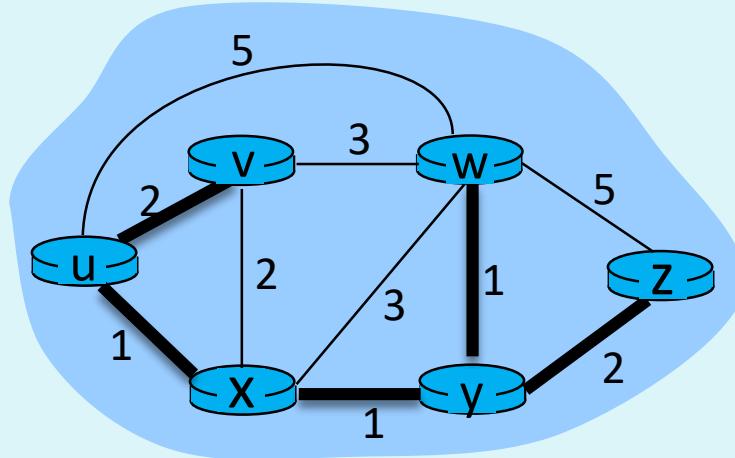
- naloga: **konfigurirajo** posredovalne tabele v usmerjevalnikih, da s tem omogočijo vzpostavitev **najcenejše poti** od začetnega do končnega usmerjevalnika
 - kaj je "cena"? Npr.: razdalja, denar, hitrost, politika, ...
- V grafu obstaja več poti med začetnim in ciljnim usmerjevalnikom.



- $c(x,x') = \text{cena povezave } (x,x')$
- $\text{cena poti } (x_1, x_2, x_3, \dots, x_p) = c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$
- usmerjevalni algoritem = algoritem, ki najde **najcenejšo pot**

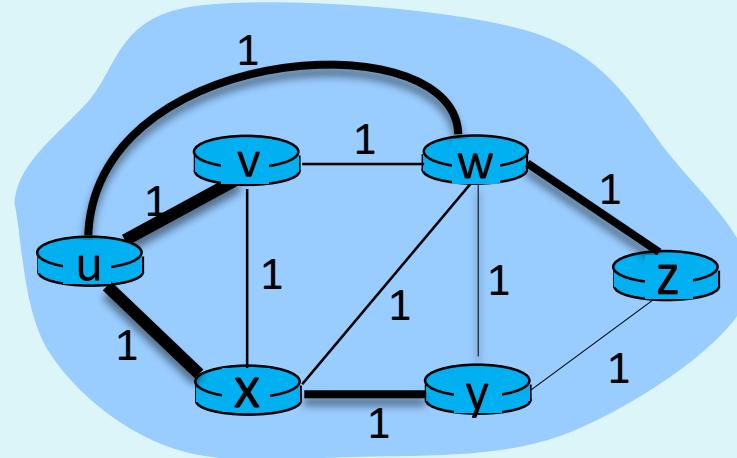
Kriteriji pri usmerjanju

- Pred optimizacijo je potrebno določiti, kaj je CENA povezave: možno je uporabiti različne kriterije: čas, ceno, število skokov ...



Usmerjanje glede na **ceno**

cilj	povezava	cena
v	(u,v)	2
x	(u,x)	1
y	(u,x)	2
w	(u,x)	3
z	(u,x)	4



Usmerjanje glede na **število skokov**

cilj	povezava	cena
v	(u,v)	1
x	(u,x)	1
y	(u,x)	2
w	(u,w)	1
z	(u,w)	2

Vrste usmerjevalnih algoritmov

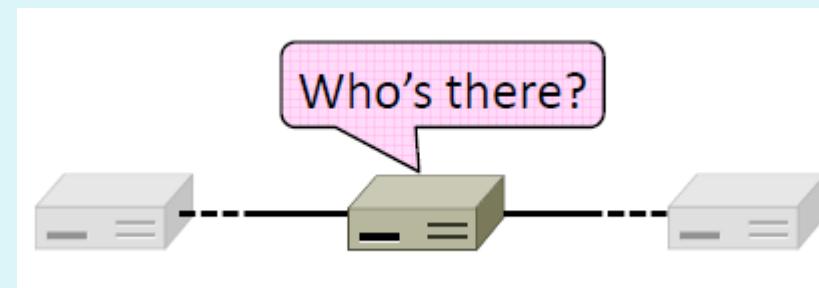
Možne so različne kombinacije lastnosti:

1. centralizirani (globalni) ali decentralizirani (porazdeljeni)

- centralizirani imajo dostopne podatke o stanju povezav v celiem omrežju (link state algoritmi)
- decentralizirani imajo dostopne podatke samo o neposredno priključenih povezavah, izračun optimalne poti poteka iterativno na osnovi vektorja razdalj (distance vector algoritmi)

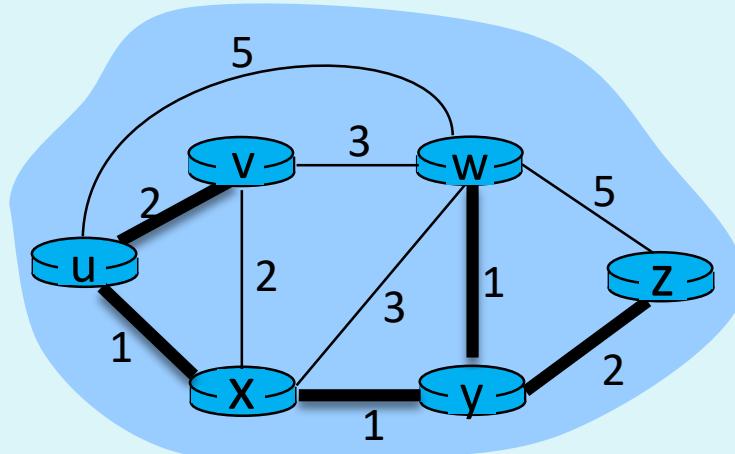
2. prilagodljivi in neprilagodljivi na obremenitev povezav

- prilagodljivi avtomatsko prilagajajo cene povezav glede na zasičenost povezave, s čimer dobijo manjšo ceno bolj proste poti



Centralizirani (globalni) algoritmi

- možna uporaba **centralnega vozlišča**, ki koordinira usmerjanje ali neodvisno izračunavanje posameznih vozlišč (**link-state**)
- uporabljajo podatke o stanju povezav v celiem omrežju (vsako vozlišče sporoča stanje povezav *vsem* ostalim vozliščem)
- vsako vozlišče zase izračuna drevo najkrajših poti do ostalih vozlišč (algoritem Dijkstra): rezultat je posredovalna tabela za to vozlišče

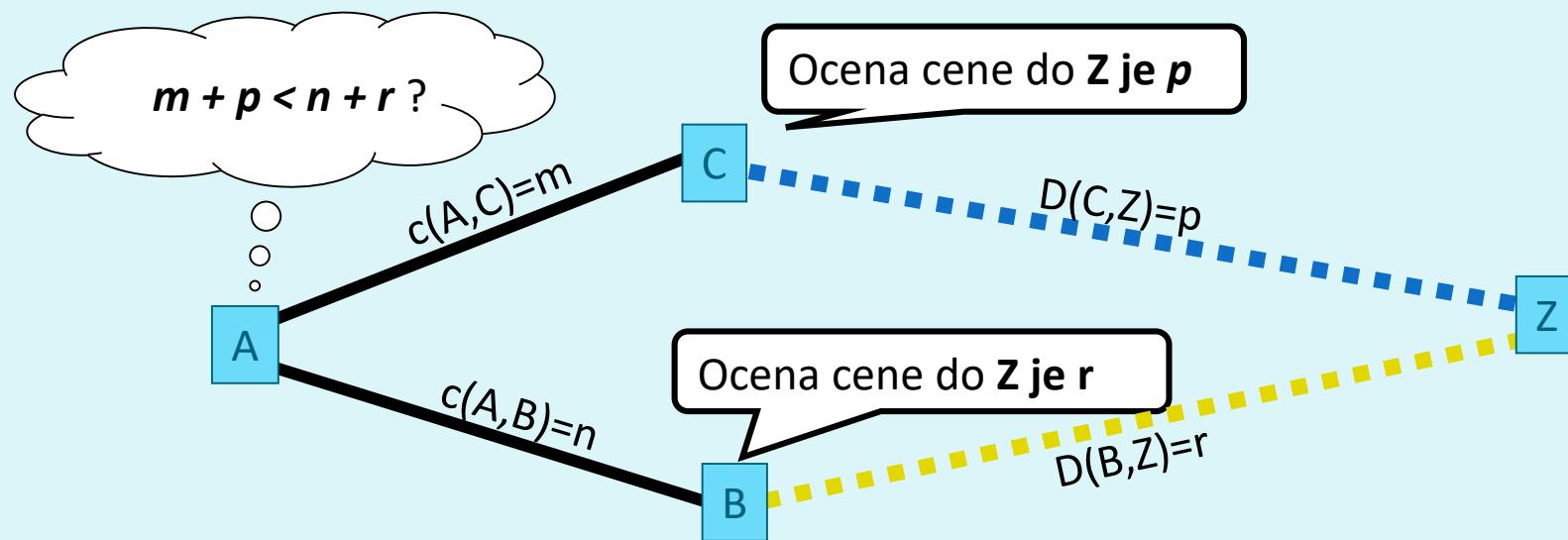


Izračunana posredovalna tabela **vozlišča u**

cilj	povezava/vrata	cena
v	(u,v)	2
x	(u,x)	1
y	(u,x)	2
w	(u,x)	3
z	(u,x)	4

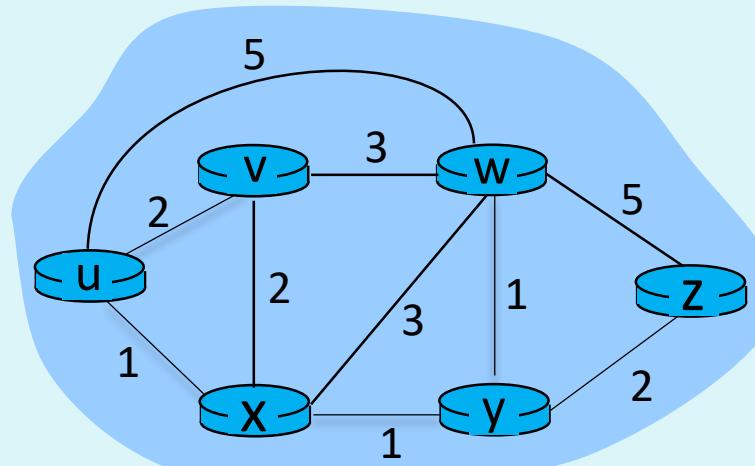
Decentralizirano (porazdeljeno) usmerjanje

- usmerjanje z **vektorjem razdalj** (angl. *distance vector*)
- vsako vozlišče izračunava posredovalno tabelo na osnovi **lokalnih podatkov**, prejetih od neposredno povezanih sosedov,
- usmerjanje je **iterativno**, računanje tabel poteka v korakih,
- vozlišče s sosedi $S=\{s_1, \dots, s_k\}$ potrebuje za izračun najcenejše poti naslednje podatke:
 - **znana cena povezave** od izvora x do sosedov s : $c(x,s)$
 - **ocena cene najcenejše poti** od sosedja s do cilja y : $D(s,y)$

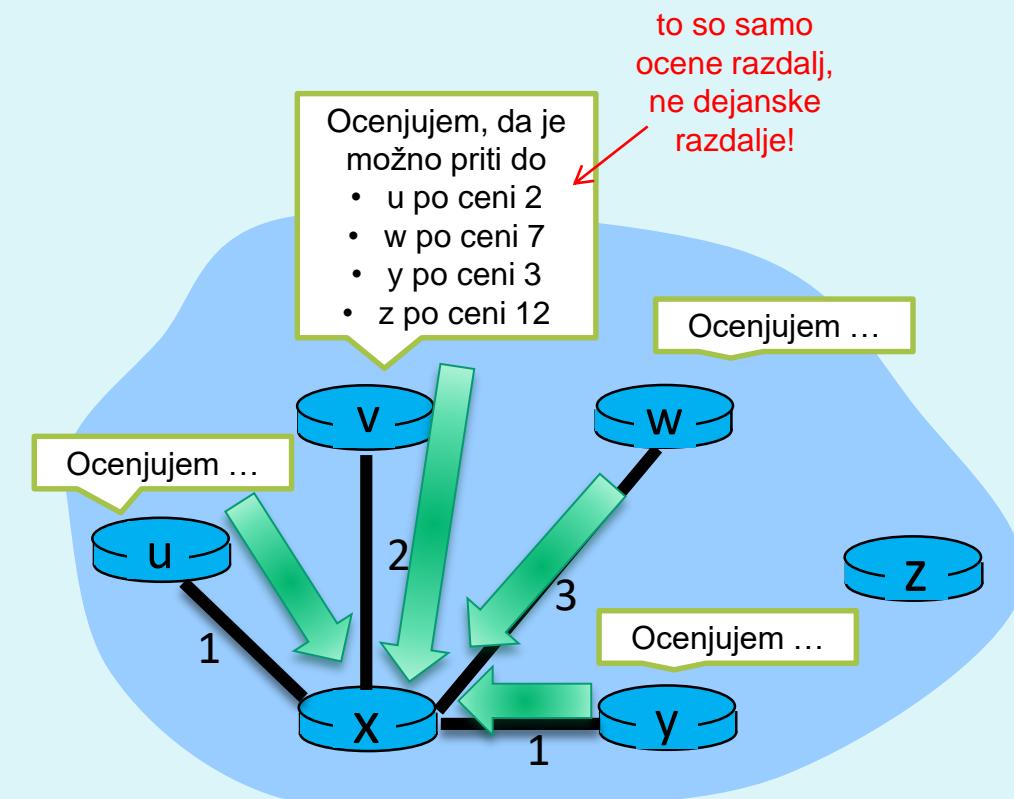


Porazdeljeno usmerjanje

- podano je omrežje:



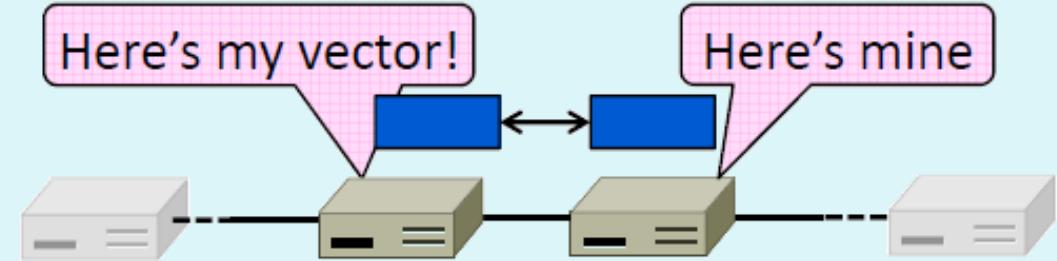
- porazdeljeno usmerjanje s stališča vozlišča x:



Usmerjanje z vektorjem razdalj

vsako vozlišče hrani podatke:

- svoj vektor razdalj
- vektorje razdalj vseh svojih sosedov
- posredovalno tabelo

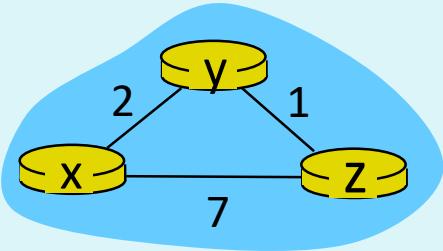


osnovna ideja:

- ko vozlišče prejme vektor razdalj od sosedja, izračuna svoj novi vektor razdalj:
$$\mathbf{D}(x, y) \leftarrow \min_{s \in S} \{c(x, s) + \mathbf{D}(s, y)\}$$
 za vsa vozlišča y
- vsako vozlišče občasno pošlje svoj vektor razdalj svojim sosedom, običajno takrat, ko zazna spremembo v svojem vektorju razdalj
- sčasoma posredovalne tabele vseh usmerjevalnikov konvergirajo v dejanske najmanjše cene

$$\begin{aligned}
 D_x(y) &= \min\{c(x,y) + D(y,y), c(x,z) + D(z,y)\} \\
 &= \min\{2+0, 7+1\} = 2
 \end{aligned}$$

$$\begin{aligned}
 D_x(z) &= \min\{c(x,y) + D(y,z), c(x,z) + D(z,z)\} \\
 &= \min\{2+1, 7+0\} = 3
 \end{aligned}$$



		cena do			cena do			cena do		
		vozlišče x			vozlišče y			vozlišče z		
		x	y	z	x	y	z	x	y	z
po	x	0	2	7	0	2	3	0	2	3
	y	∞	∞	∞	2	0	1	2	0	1
	z	∞	∞	∞	7	1	0	3	1	0

		cena do			cena do			cena do		
		vozlišče y			vozlišče z			vozlišče x		
		x	y	z	x	y	z	x	y	z
po	x	∞	∞	∞	0	2	7	0	2	3
	y	2	0	1	2	0	1	2	0	1
	z	∞	∞	∞	7	1	0	3	1	0

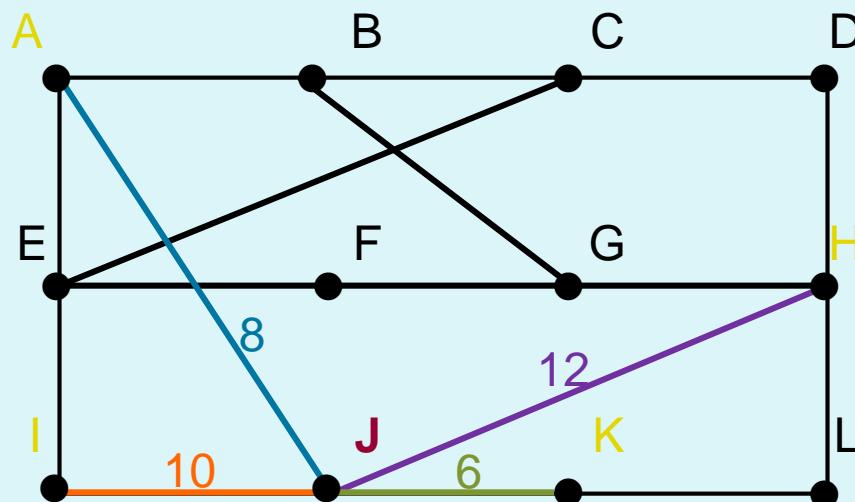
		cena do			cena do			cena do		
		vozlišče z			vozlišče x			vozlišče y		
		x	y	z	x	y	z	x	y	z
po	x	∞	∞	∞	0	2	7	0	2	3
	y	∞	∞	∞	2	0	1	2	0	1
	z	7	1	0	3	1	0	3	1	0

→ čas

The diagram shows three tables representing the shortest path tree from node x. The first table is for node x, the second for node y, and the third for node z. The columns represent the nodes x, y, and z. The rows represent the nodes x, y, and z again. The values in the cells indicate the shortest distance from the root node x to each node. Red ovals highlight the values 0, 2, 7, 3, 1, and 0 in the first table, corresponding to the edges (x,y), (x,z), (y,z), (x,y), (y,z), and (z,y) respectively. Green arrows point from the highlighted values in the first table to the corresponding values in the second and third tables, indicating the propagation of information through the network.

Porazdeljeno usmerjanje: naloga

J dobi tabele od sosedov. Poišči njegovo novo posredovalno tabelo!



JA: 8 JH: 12

JI: 10 JK: 6

prejeti vektorji razdalj

Dobi od Smer	A	I	H	K
A	-	24	20	21
B	12	36	31	28
C	25	18	19	26
D	40	27	8	24
E	14	7	30	22
F	23	20	19	40
G	18	31	6	31
H	17	20	-	19
I	21	-	14	22
J	9	11	7	10
K	24	22	22	-
L	29	33	9	9

nova posredovalna tabela

Smer	Ocena	Sosed
A	8	A
B	20	A
C	28	I
D	20	H
E	17	I
F	30	I
G	18	H
H	12	H
I	10	I
J	-	-
K	6	K
L	15	K

Porazdeljeno usmerjanje

- usmerjevalne tabele se prilagodijo na spremembe v cenah povezav
- dva principa:
 - "*good news travel fast*": podatek o znižanju cene povezav se hitro razširi in posredovalne tabele se hitro ustrezno prilagodijo
 - "*bad news travel slow*": podatek o povišanju cene povezav se širi počasi, lahko povzroči t. i. "štetje do neskončnosti" (*count to infinity problem*)

vzpostavitev nove povezave A-B,
odkrivanje nove poti od B/C/D/E do A



	A	B	C	D	E
začetek	∞	∞	∞	∞	
1. iteracija	1	∞	∞	∞	
2. iteracija	1	2	∞	∞	
3. iteracija	1	2	3	∞	
4. iteracija	1	2	3	4	

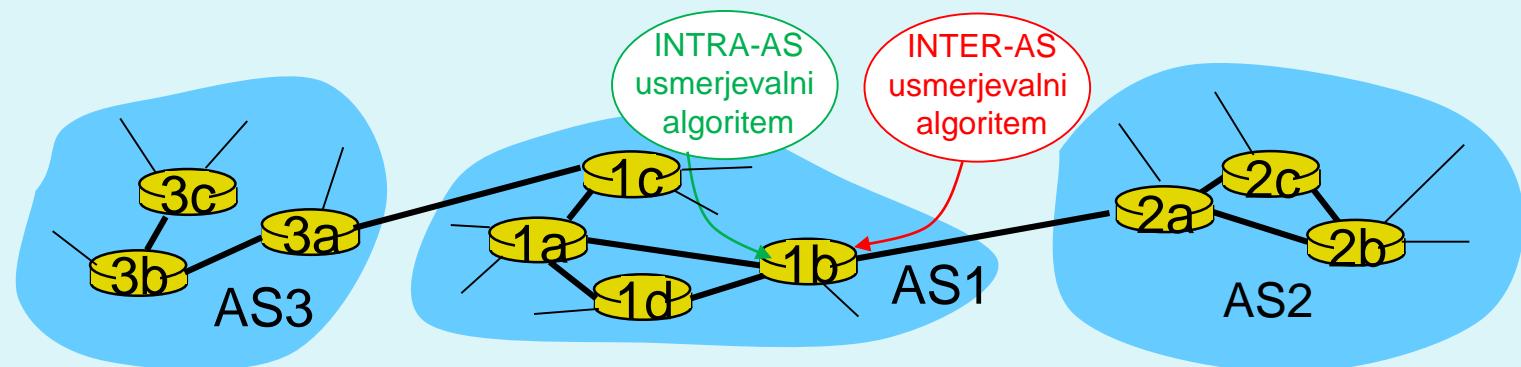
prekinitev povezave A-B



	A	B	C	D	E
začetek	1	2	3	4	
1. iteracija	3	2	3	4	
2. iteracija	3	4	3	4	
3. iteracija	5	4	5	4	
4. iteracija	5	6	5	6	
5. iteracija	7	6	7	6	
		...			

Hierarhično usmerjanje

- če bi imeli vse usmerjevalnike v istem omrežju, bi imeli težave:
 - velike usmerjevalne tabele (ne bi mogli shraniti vseh destinacij)
 - administrator vsakega usmerjevalnega omrežja želi administrirati omrežje po svoje
- rešitev:
 - skupine usmerjevalnikov organiziramo v **avtonomne sisteme (AS)**, ki so pod neodvisnimi administracijami
 - usmerjevalni protokoli v enem AS so enaki (vse uporabljajo **isti usmerjevalni protokol (INTRA-AS usmerjevalni algoritem)**, npr. distance-vector ali link-state), različni AS pa lahko **različne**
 - za povezovanje AS med seboj se uporablja **INTER-AS usmerjevalni protokol**, ki pa mora biti v celotnem omrežju **enak!** Z njim se usmerjevalniki naučijo, kako iz enega AS usmeriti pakete v destinacije, ki so v drugih AS



Usmerjanje v Internetu

- **Intra-AS usmerjanje:** zanj skrbijo IGP (Interior Gateway Protocols), primeri:
 - **RIP: Routing Information Protocol (se opušča)**
 - usmerjanje z **vektorjem razdalj** (distance vector), algoritmom **optimizira število hopov**, cena vsake povezave je torej 1; največja dovoljena cena je 15
 - vektor razdalj se **razpošilja na 30 s**; če se ne razpošlje v 180 s, se povezava smatra za prekinjeno
 - **OSPF: Open Shortest Path First**
 - usmerjanje glede na **stanje povezav** (link state)
 - obvestila se s poplavljajem posredujejo celotnemu sistemu, ki preračuna najkrajše poti
 - prednosti: varnost, usmerjanje po več poteh, razpošiljanje, hierarhično usmerjanje
 - **IGRP: Interior Gateway Routing Protocol**
 - Cisco-va izboljšava protokola RIP, usmerjanje z vektorjem razdalj
 - cena se izračuna kot utežena vsota pasovne širine, zakasnitve, obremenitve, MTU in zanesljivosti
- **Inter-AS usmerjanje:** zanj skrbi BGP (Border Gateway Protocol, BGP4)
 - omogoča, da omrežja oglašujejo svojo prisotnost drugim omrežjem

Naslednje poglavje?

- transportna plast!

