

# Računalniške komunikacije

2020/21

omrežna plast  
ICMP, IPv6, usmerjanje

# Pridobljeno znanje s prejšnjih predavanj

- **omrežna plast**

- **storitve** omrežne plasti (zagotovljena dostava, čas, zaporedje, pasovna širina, varianca zakasnitve, varnost)
- **modeli storitev** (best effort, CBR, ABR)
- vrste omrežij:
  - **povezavna**: usmerjanje na osnovi **identifikatorjev navideznih vodov**
  - **nepovezavna**: usmerjanje na osnovi **naslovov prejemnika**
- **posredovalne tabele**: ujemanje najdaljše predpone
- **protokol IPv4**:
  - splošna polja
  - fragmentacija
  - naslavljanje (32-bitni naslovi, podomrežja, hierarhija naslovnega prostora, razredi/CIDR)
  - dinamično dodeljevanje naslovov (DHCP)
  - prevajanje omrežnih naslovov (NAT) (varčevanje naslovnega prostora)

# Prednosti in slabosti pristopa NAT

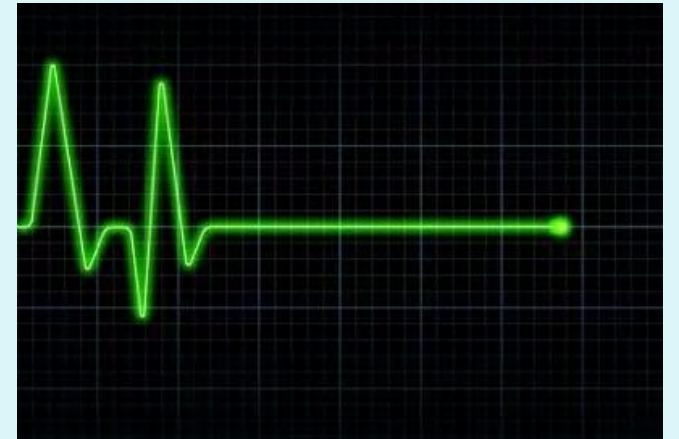
## PREDNOSTI

- zadošča samo 1 javni naslov za dostop celega omrežja do Interneta
- naslove notranjih naprav in ponudnika interneta (!) lahko **spreminjamo 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

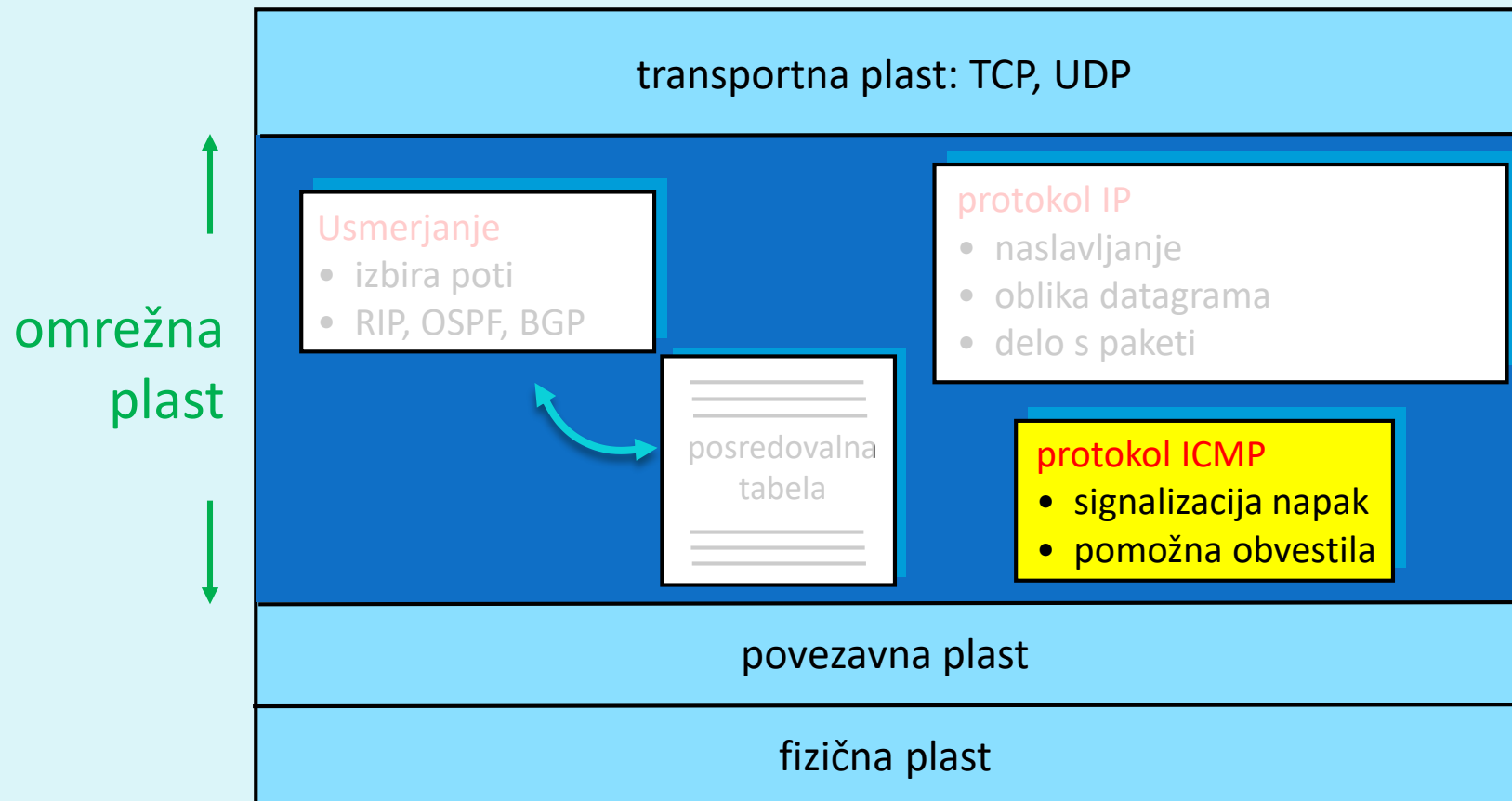
## KRITIKA

- usmerjevalniki **naj bi delali na 3. plasti** (torej ne bi imeli opravka z vrati - ki so del 4. plasti!)
- vrata (porti) so namenjeni **naslavljanju 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)



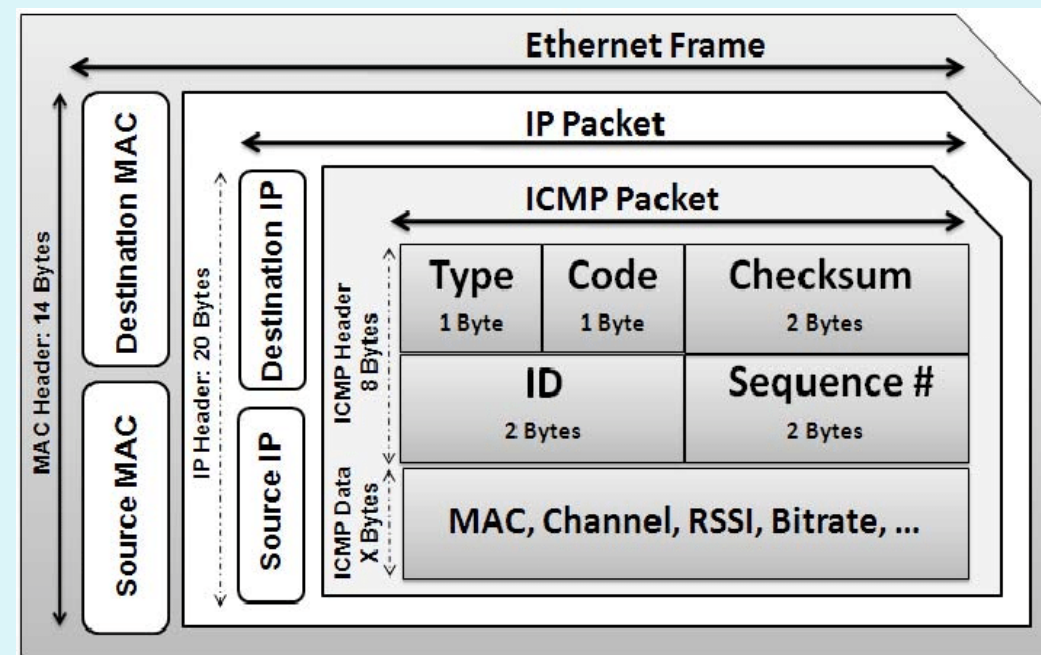
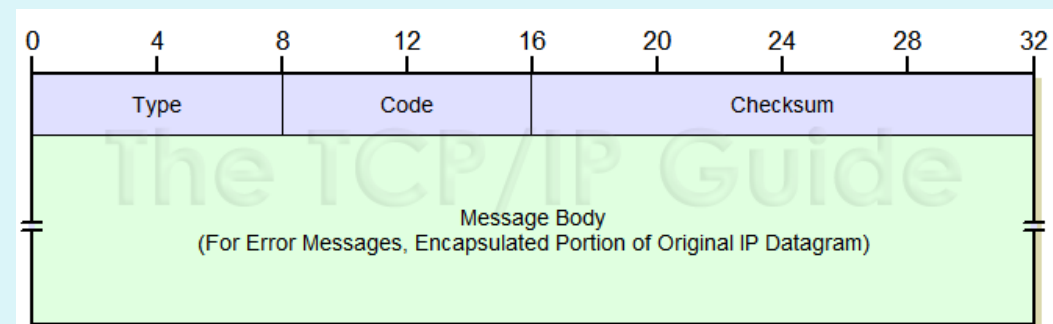
# Funkcije omrežne plasti



# 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



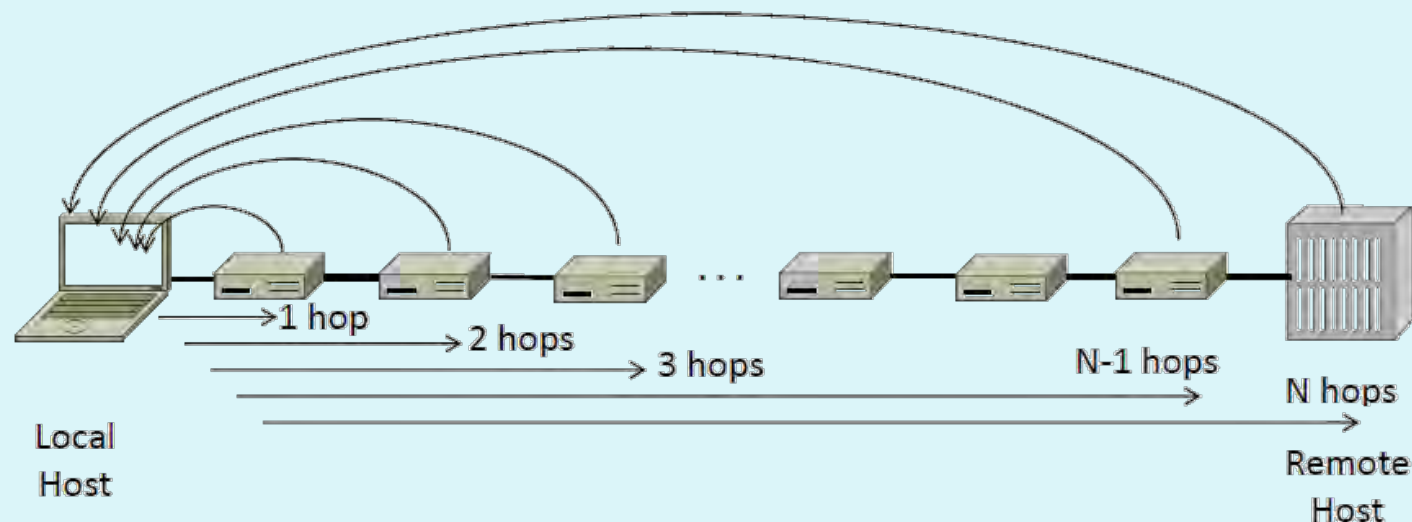
# Traceroute: aplikacija, ki uporablja ICMP

- **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



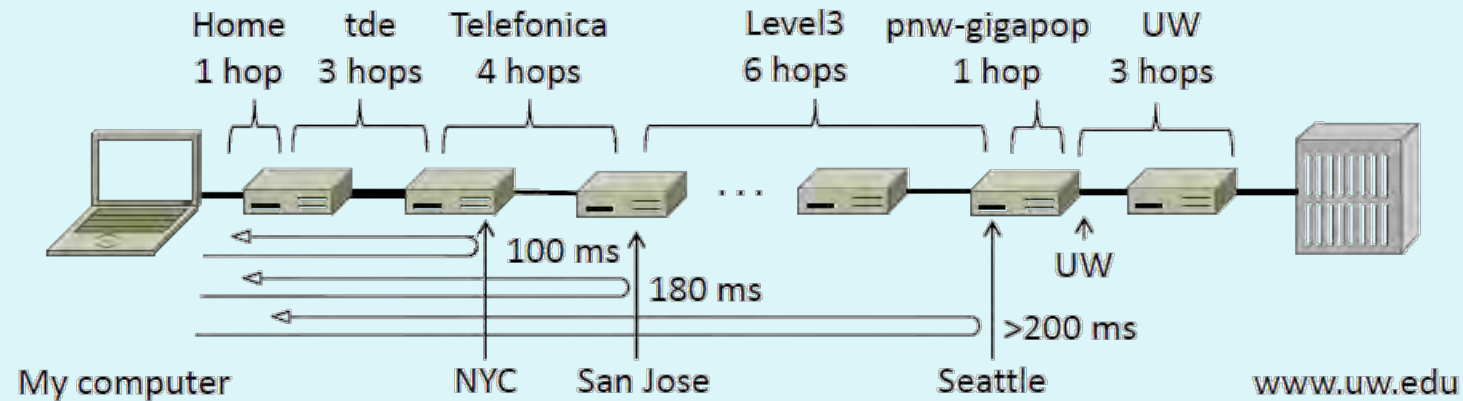
# Delovanje aplikacije Traceroute

- 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



```
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:
  0  <1 ms    <1 ms    <1 ms    Broadcom.amis.lan [192.168.1.254]
  1  1 ms     1 ms     1 ms     asr-lj.amis.net [212.18.32.174]
  2  1 ms     1 ms     1 ms     nx-lj1-te-0-3-0.amis.net [212.18.44.161]
  3  8 ms     8 ms     8 ms     nx-vil-te-0-0-0.amis.net [212.18.44.142]
  4  23 ms    23 ms    23 ms    30gigabitethernet4-3.core1.fra1.he.net [80.81.192.172]
  5  32 ms    33 ms    40 ms    10gigabitethernet1-4.core1.par2.he.net [184.105.213.162]
  6  122 ms   110 ms   121 ms   10gigabitethernet7-1.core1.ash1.he.net [184.105.213.93]
  7  181 ms   180 ms   187 ms   10gigabitethernet11-1.core1.pao1.he.net [184.105.213.177]
  8  186 ms   181 ms   181 ms   stanford-university.10gigabitethernet1-4.core1.pao1.he.net [216.218.209.118]
  9  187 ms   187 ms   187 ms   boundarya-rtr.Stanford.EDU [68.65.168.33]
 10  *        *        *        Request timed out.
 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 ( $3.4 \times 10^{38}$ ) oziroma  $6.65 \times 10^{23}$  naslovov na m<sup>2</sup> 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:

```
0010000111011010000000001101001100000000000000000010111100111011
00000010101010100000000011111111111111110001010001001110001011010
```

v teh (zadnjih) 32 bitov  
gre IPv4 naslov, če to  
želimo zaradi  
kompatibilnosti

- razdeljen na osem 16-bitnih skupin:

```
0010000111011010 0000000011010011 0000000000000000 0010111100111011
0000001010101010 0000000011111111 1111111000101000 1001110001011010
```

- zapisan šestnajstiško, ločeno z dvopičji

21da:00d3:0000:2f3b:02aa:00ff:fe28:9c5a

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

samo eno skupino blokov lahko pokrajšamo, če jih imamo  
več, potem pokrajšamo enega izmed teh (če bi pokrajšali  
oba, ne bi vedeli, koliko je vsak dolg)

- 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

zadnjih 32 bitov naslova nadomestimo z IPv4 naslovom

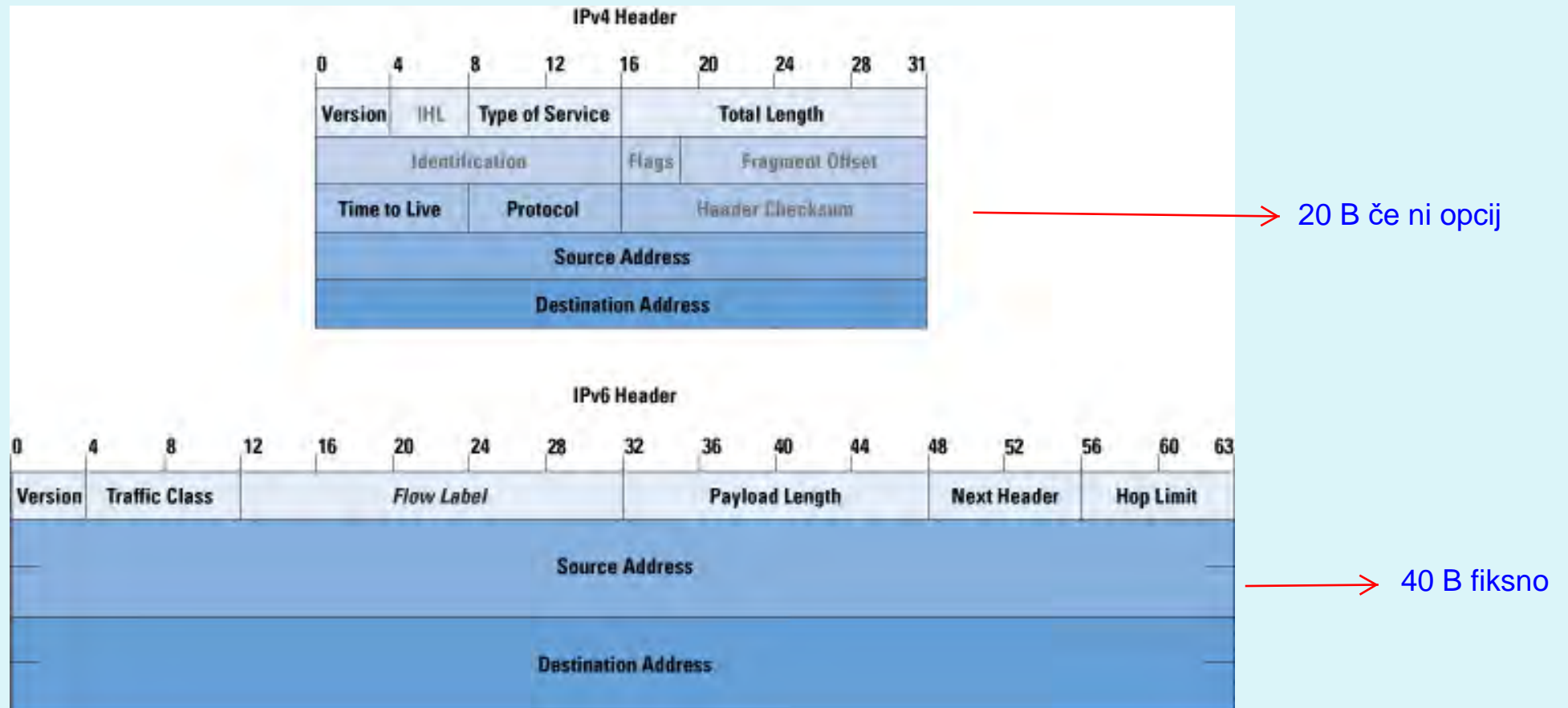
- 193.2.72.1 → ::193.2.72.1 (lahko celo pustimo pike iz IPv4 naslova!)

# IPv6: hitrejšje procesiranje paketov

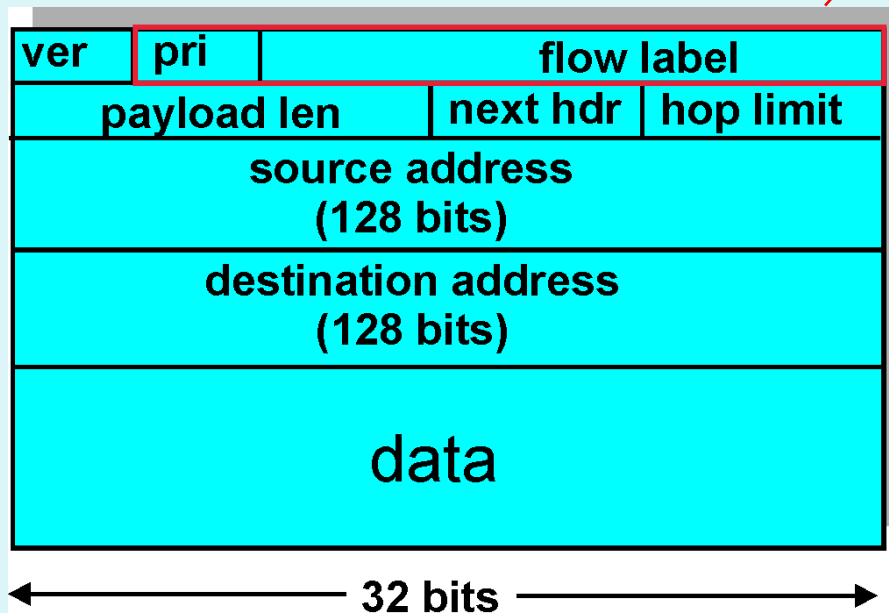
- **fragmentacija se ne izvaja več** (za delitev paketov skrbita pošiljatelj in sprejemnik). V primeru, da je paket prevelik, usmerjevalnik zavrže paket in sporoči pošiljatelju "Packet Too Big",
  - nova verzija protokola ICMP: **ICMPv6** (eno od sporočil tega protokola je zgornji "Packet Too Big"),
- 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



# IPv6 format paketa



obe polji za kvaliteto prenosa

oboje se  
ni prijelo

polje za multipleksiranje

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

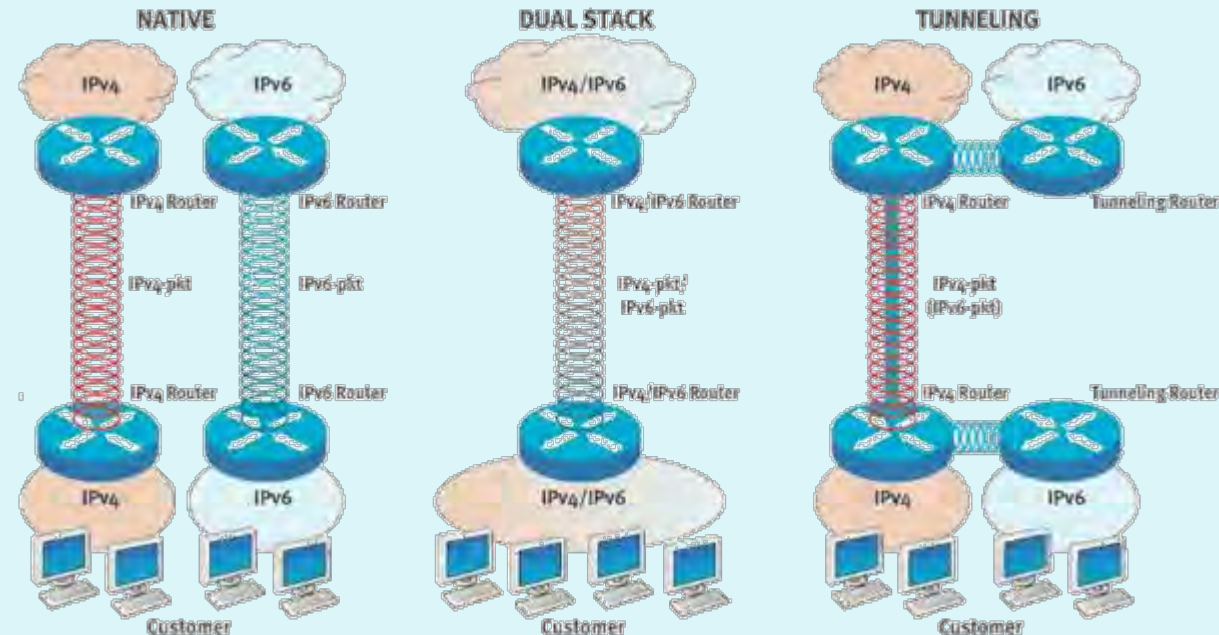


- **VER** (4b): verzija IP protokola (6)
- **PRI** ali **TRAFFIC CLASS** (8b): podobno kot Type Of Service pri IPv4, oznaka prioritete za posebne pakete / aplikacije
- **FLOW LABEL** (20b): oznaka "toka" podatkov, ki omogoči posebno zagotavljanje kakovosti storitve (npr. audio/video)
- **PAYLOAD LENGTH**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



# Prehod iz IPv4 -> IPv6

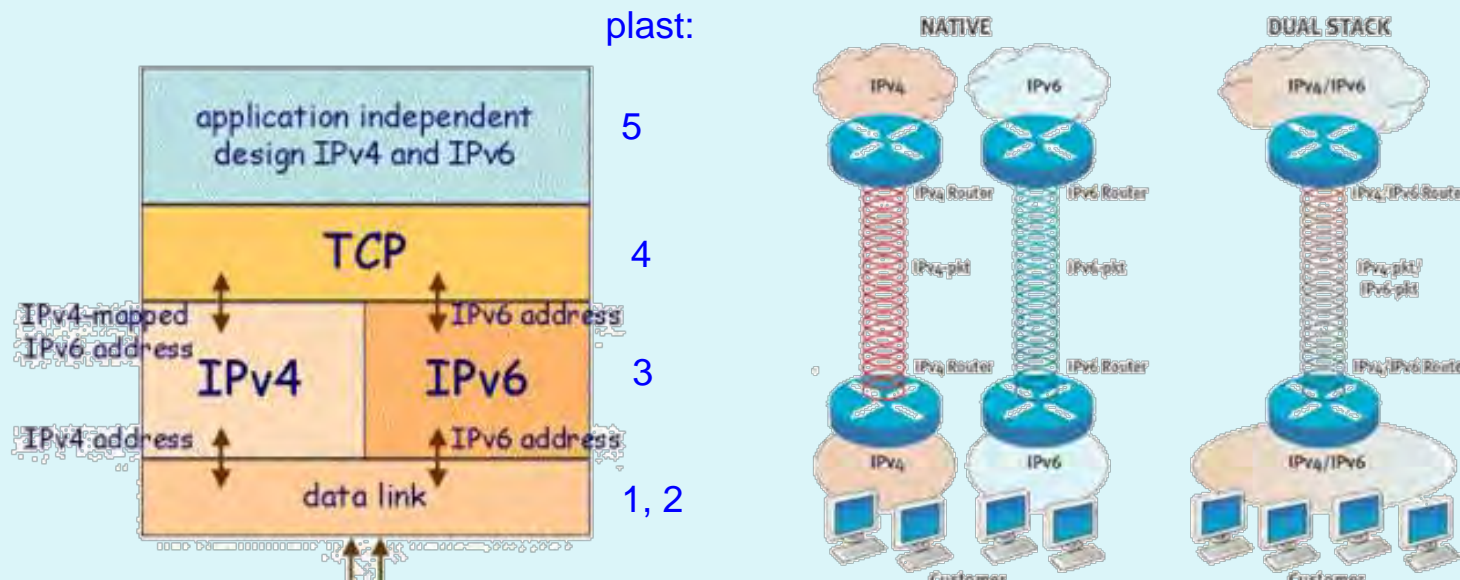
- vseh naprav ni mogoče nadgraditi naenkrat ("flag day")
- za prehod se najpogosteje uporabljata dve rešitvi:
  - dvojni sklad (*dual-stack*, vozlišča uporabljajo vzporedni implementaciji IPv4 in IPv6)
  - tuneliranje (*tunneling*, paket IPv6 zapakiramo v paket IPv4 kot podatke)





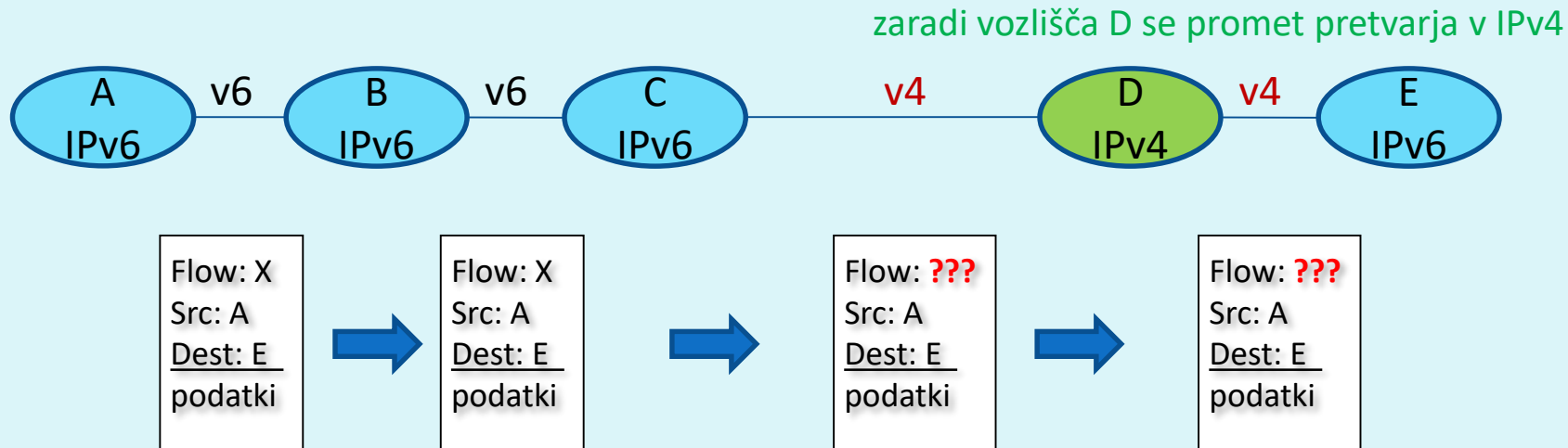
# Prehodni mehanizem: Dvojni sklad

- usmerjevalnik "govori" **IPv4 in IPv6**. Z vozlišči, ki uporabljajo 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

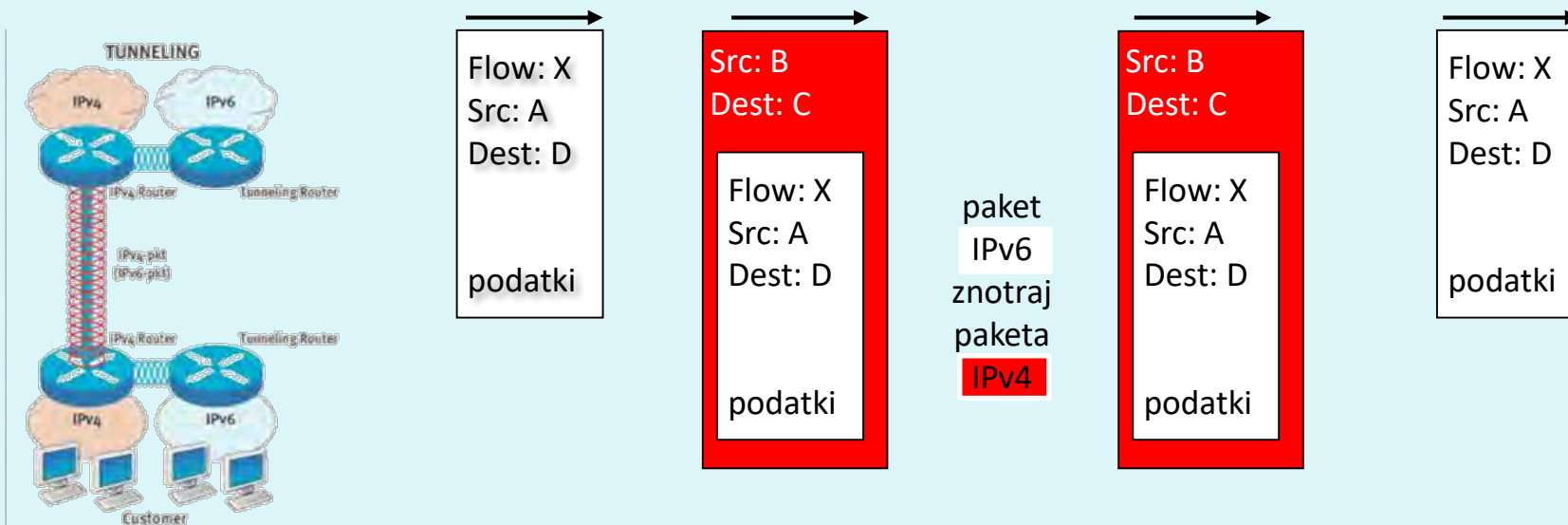
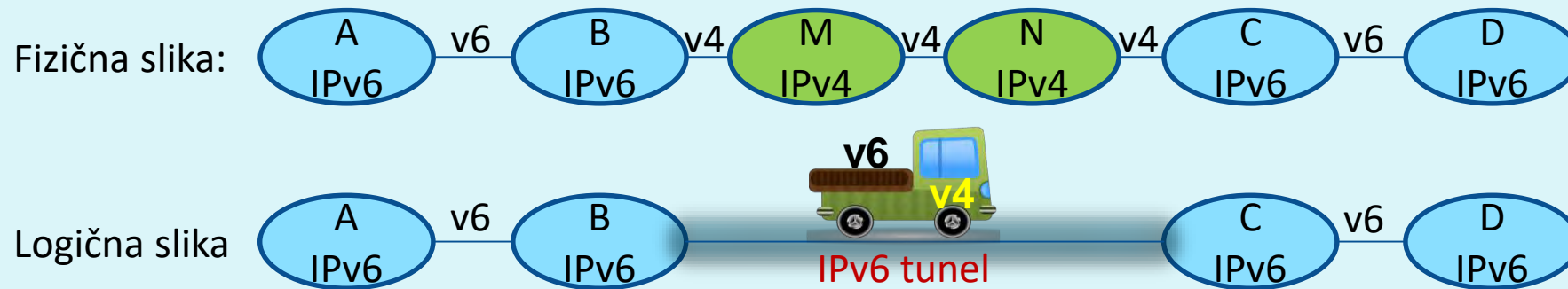
- če je na poti med dvema IPv6 vozliščema kakšno IPv4, se bo promet vmes **pretvarjal v IPv4**; Specifična polja protokola IPv6 (flow label?) se bodo izgubila!



če bi bil tu še F, ki pozna IPv6, lahko E pretvori paket v IPv6, ampak ne more obnoviti flow labela

# 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 algoritmi in ključih
  - enkripcija in dekripcija (zakrivanje komunikacije)
  - integriteta podatkov
  - avtentikacija izvora

Authentication Header

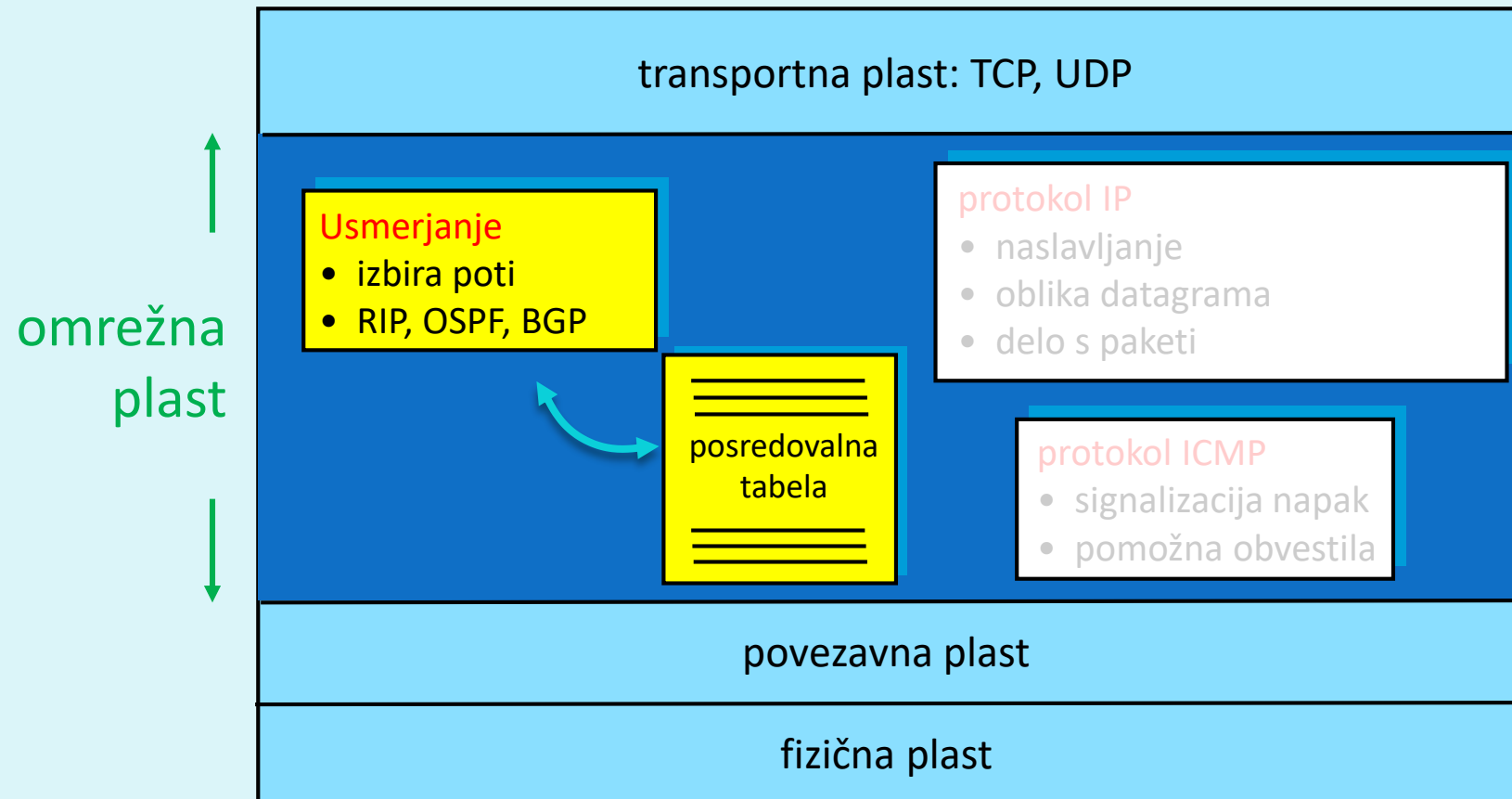
Encapsulating  
Security Payload



# Usmerjanje

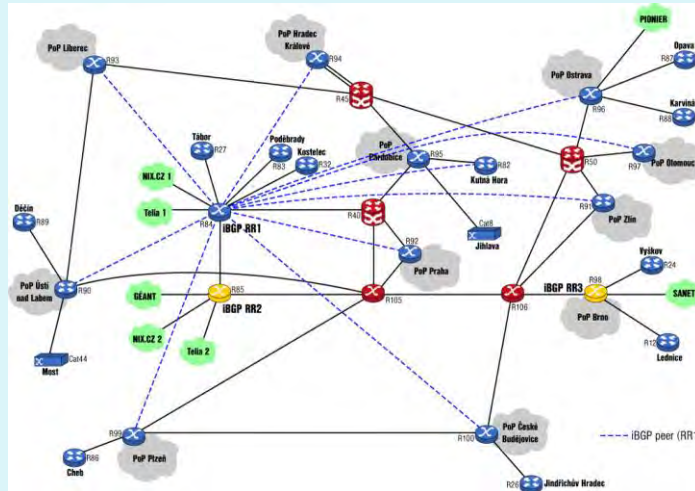


# Funkcije omrežne plasti



# Usmerjanje

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

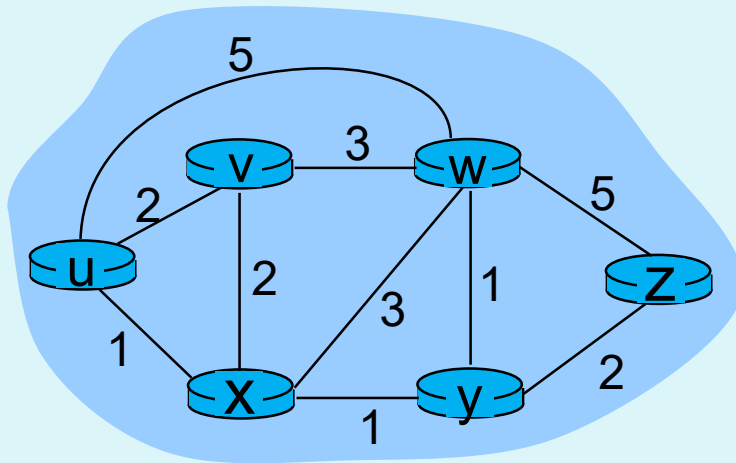


- omrežja modeliramo s teorijo grafov  $G = \langle V, P \rangle$ :
  - vozlišča  $V$ : usmerjevalniki
  - povezave  $P$ : komunikacijske povezave



# 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.

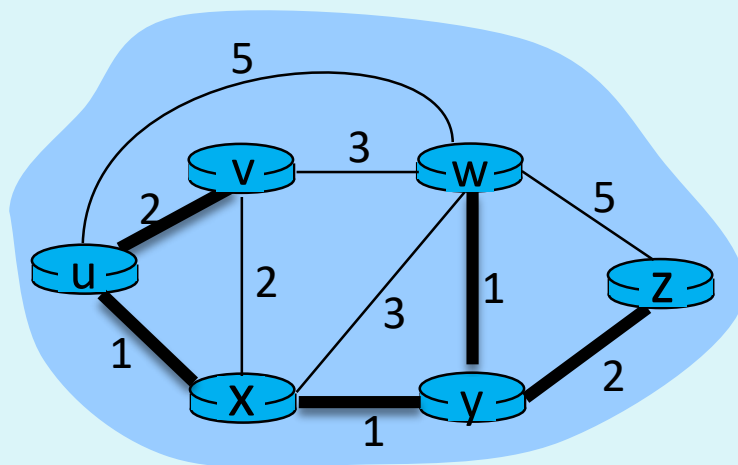


- $c(x, x') = \text{cena povezave } (x, x')$
- **cena poti**  $(x_1, x_2, x_3, \dots, x_p) = c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$
- usmerjevalni algoritem = algoritem, ki najde **najcenejšo pot**



# Kriteriji pri usmerjanju

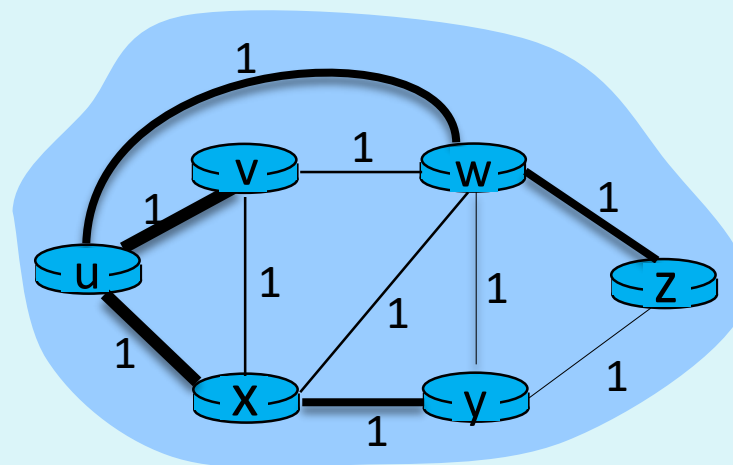
- Pred optimizacijo je potrebno določiti, kaj je CENA povezave: možno je uporabiti različne kriterije: čas, ceno, število skokov ...



Usmerjanje glede na **ceno**

v primeru gre  
vse iz u

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



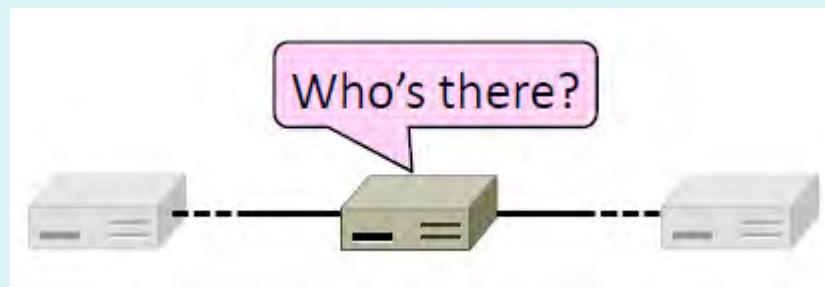
Usmerjanje glede na **število skokov**

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:

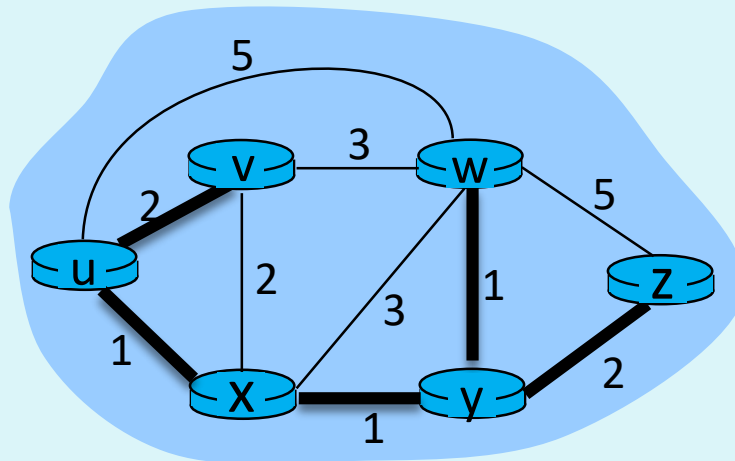
1. **centralizirani** (globalni) ali **decentralizirani** (porazdeljeni)
  - centralizirani imajo dostopne podatke o stanju povezav v celem omrežju (*link state* algoritmi)
  - decentralizirani imajo dostopne podatke samo o neposredno priključenih povezavah, izračun optimalne poti poteka iterativno na 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**) *usmerjajo na podlagi stanj povezav*
- uporabljajo podatke o stanju povezav v celem 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

*periodično vozlišča računajo tabele na novo zaradi sprememb v omrežju itd.*

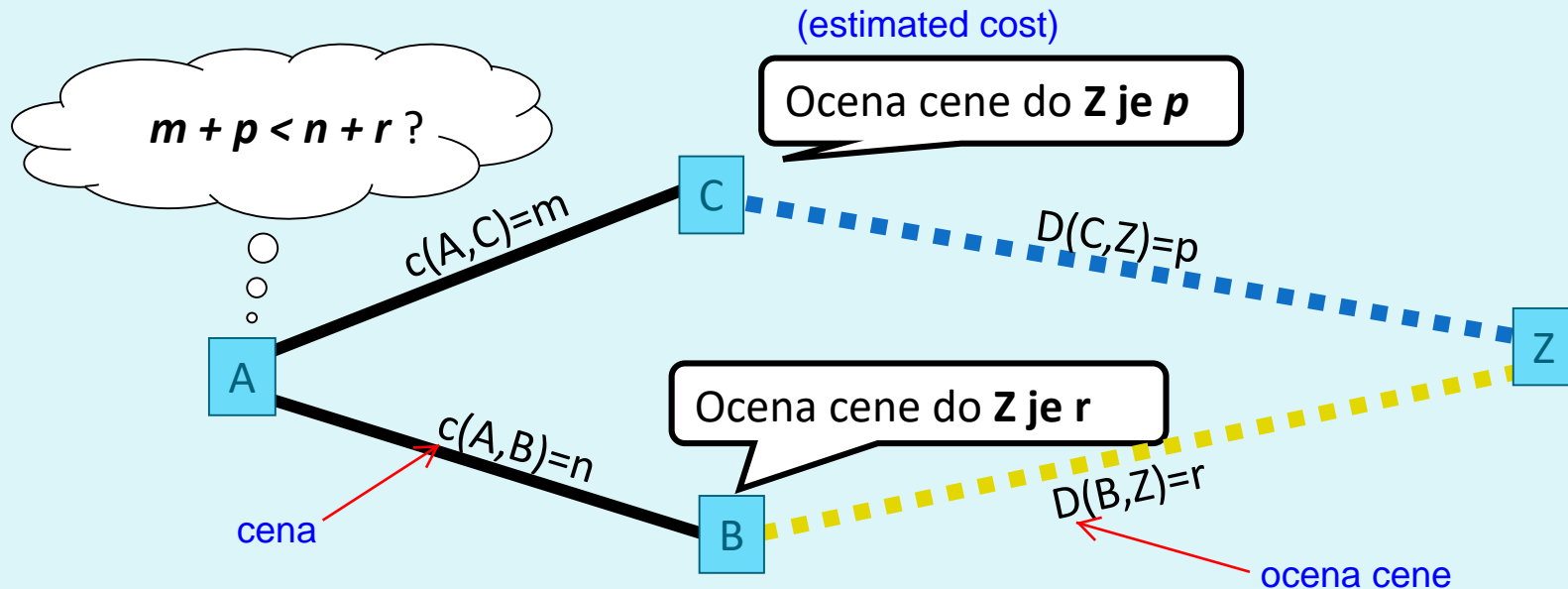


Izračunana posredovalna tabela **vozlišča u**

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

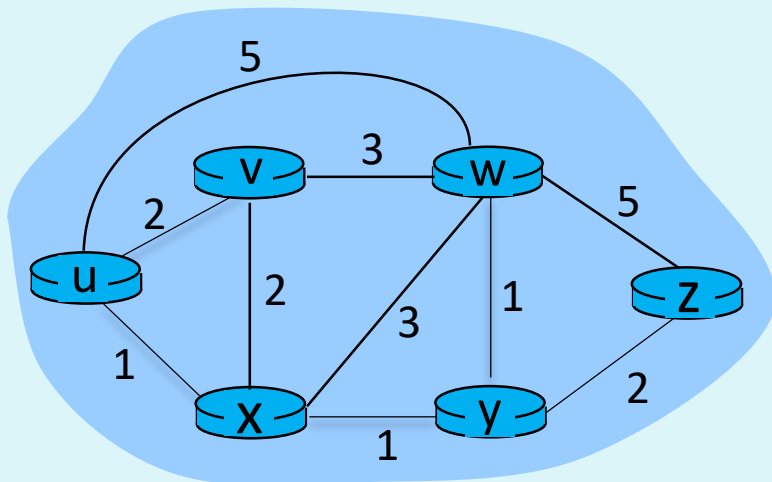
# Decentralizirano (porazdeljeno) usmerjanje

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

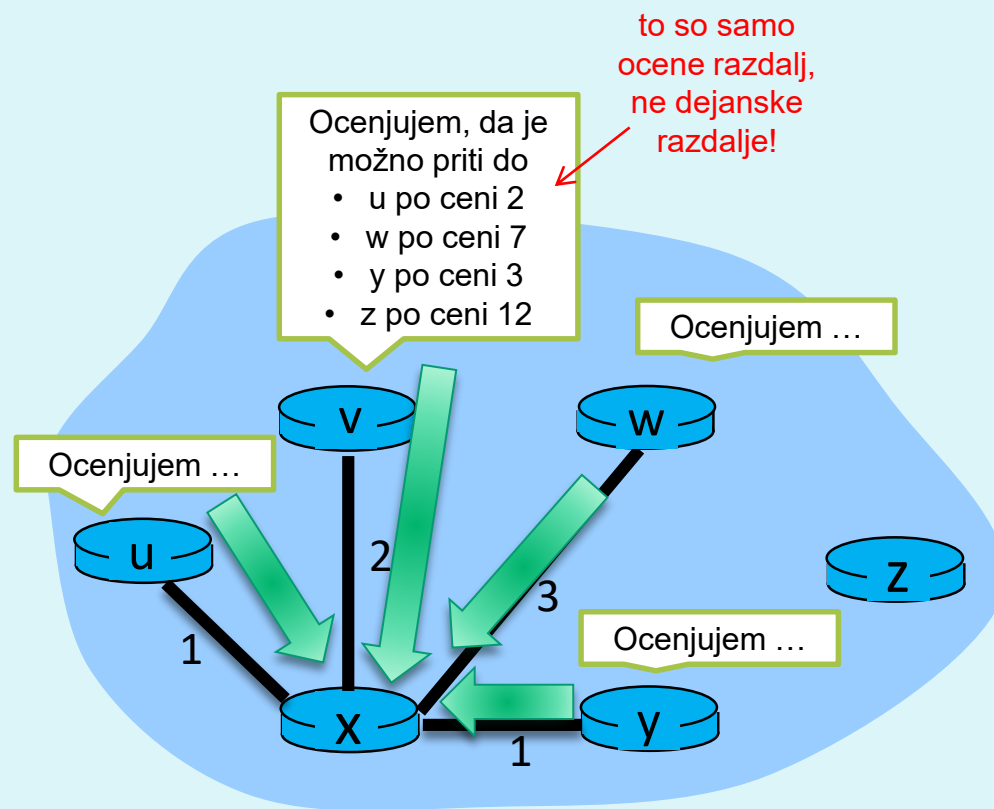


# Porazdeljeno usmerjanje

- podano je omrežje:



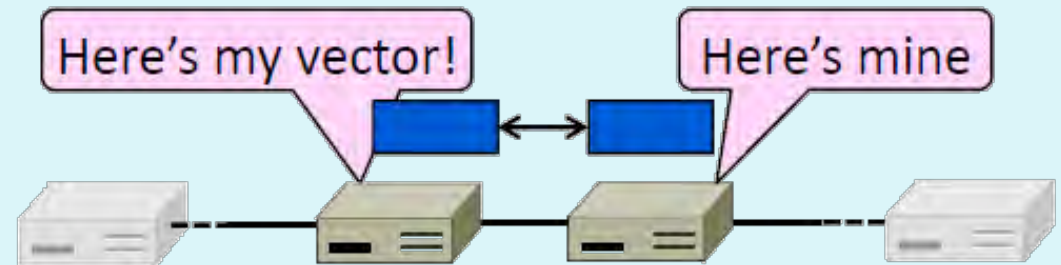
- porazdeljeno usmerjanje s stališča vozlišča x:



# Usmerjanje z vektorjem razdalj

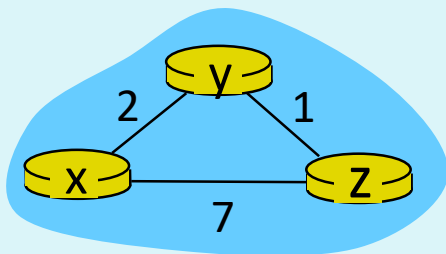
vsako vozlišče hrani podatke:

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



osnovna ideja:

- ko vozlišče prejme vektor razdalj od sosedu, izračuna svoj novi vektor razdalj:  
$$D(x, y) \leftarrow \min_{s \in S} \{c(x, s) + D(s, y)\}$$
 za vsa vozlišča  $y$  to izračuna za vsakega sosedu in vzame minimum
- 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  
= se naučijo stanja v omrežju



1. korak: vozlišča vejo samo koliko potrebujejo do svojih sosedov

2. korak: sosednja vozlišča si izmenjajo vektorje razdalj

3. korak: vozlišča preračunajo to, kar vejo

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

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

x-u je ceneje pošiljati z-ju preko y-a

**vozlišče x**

	cena do	x	y	z
od	x	0	2	7
y		$\infty$	$\infty$	$\infty$
z		$\infty$	$\infty$	$\infty$

**vozlišče y**

	cena do	x	y	z
od	x	$\infty$	$\infty$	$\infty$
y		2	0	1
z		$\infty$	$\infty$	$\infty$

**vozlišče z**

	cena do	x	y	z
od	x	$\infty$	$\infty$	$\infty$
y		$\infty$	$\infty$	$\infty$
z		7	1	0


cena do				
	x	y	z	
od	x	0	2	3
y		2	0	1
z		7	1	0

		cena do		
		x	y	z
od	x	0	2	7
	y	2	0	1
	z	7	1	0

		cena do		
		x	y	z
od	x	0	2	7
	y	2	0	1
	z	3	1	0

		cena do		
		x	y	z
od	x	0	2	3
	y	2	0	1
	z	3	1	0

		cena do		
		x	y	z
od	x	0	2	3
	y	2	0	1
	z	3	1	0

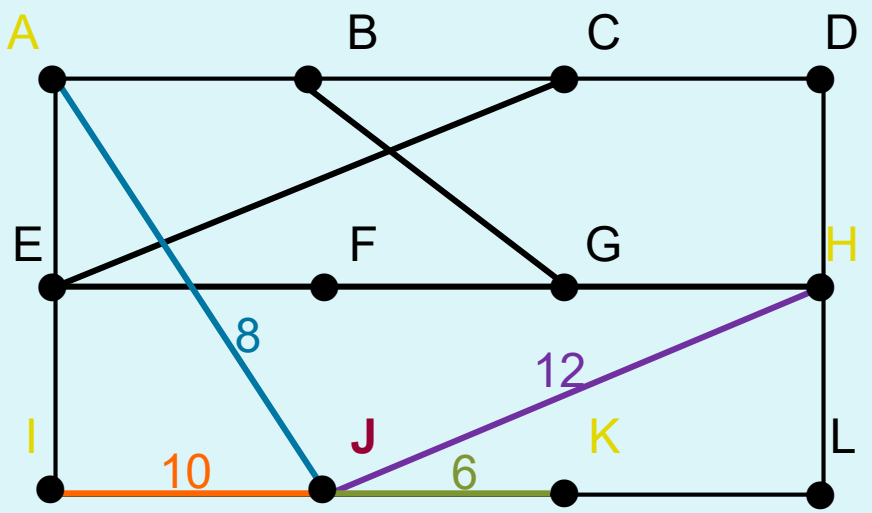


		cena do		
		x	y	z
od	x	0	2	3
	y	2	0	1
	z	3	1	0

čas

# Porazdeljeno usmerjanje: naloga

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



JA: 8    JH: 12  
JI: 10    JK: 6

Primer za ceno (J, E):  
- čez A:  $8 + 14 = 22$   
- čez H:  $12 + 30 = 42$   
- čez K:  $6 + 22 = 28$   
- čez I:  $10 + 7 = 17$

A ocenjuje, koliko stane do B, C, D itd.  
I ocenjuje, koliko stane do A, B, C itd.

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		$\infty$	$\infty$	$\infty$	$\infty$
1. iteracija	1		$\infty$	$\infty$	$\infty$
2. iteracija	1	1		$\infty$	$\infty$
3. iteracija	1	1	2		$\infty$
4. iteracija	1	1	2	3	

**prekinitev povezave A-B**

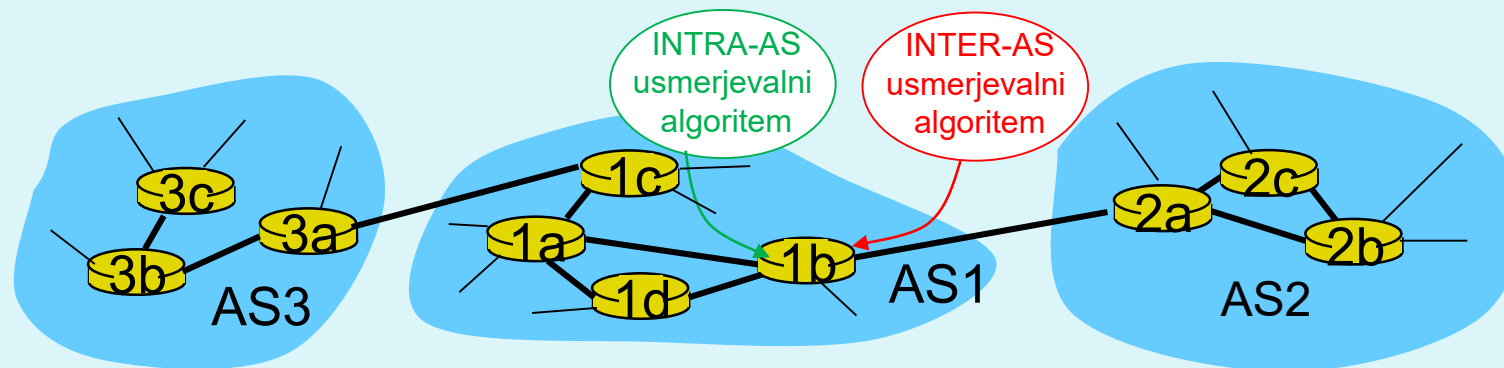


	A	B	C	D	E
začetek		1	2	3	4
1. iteracija	3		2	3	4
2. iteracija	3	4		3	4
3. iteracija	5	4	5		4
4. iteracija	5	6	5	6	
5. iteracija	7	6	7	6	6

...

# 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
  - usmerjevalniki v istem AS 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), algoritem **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 poplavljanjem 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

# Naslednjič gremo naprej!

- transportna plast!

