

Omrežna plast

Naloge omrežne plasti

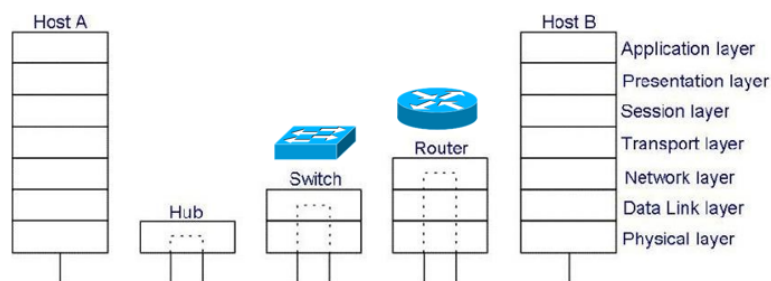
- transport segmenta od **končnega** pošiljatelja do prejemnika
- enkapsulacija segmentov transportne plasti v **pakete** na strani pošiljatelja
- prisotna v vseh omrežnih napravah in v jedru omrežja(**usmerjevalnikih**)

Enkapsulirajo se podatki transportne plasti, dekapsovirajo se podatki povezavne plasti.

Usmerjevalnik

Je naprava, ki deluje na omrežni plasti in skrbi za transport datagrama po jedru omrežja.

- povezave med različnimi mediji in protokoli
- izvajajo **usmerjanje** in **posredovanje**



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
- **usmerjanje** (routing): določitev poti paketov od izvora do cilja. Je "kolektivno delo" vseh omrežnih naprav na poti, ki izvajajo usmerjevalne algoritme (in protokole).

Storitve omrežne plasti

Omrežna plast **lahko** omogoča naslednje storitve:

- **zagotovljena dostava** paketov
- dostava paketov v **zagotovljenem času**
- dostava paketov v **pravem zaporedju**
- zagotovljena spodnja **meja pasovne širine**
- največja dovoljena varianca zakasnitve (jitter):
 $t_{pošiljanja}(P2) - t_{pošiljanja}(P1) \sim t_{prejetja}(P2) - t_{prejetja}(P1)$
- **varno komunikacijo** (zaupnost, integriteto podatkov, avtentikacijo)

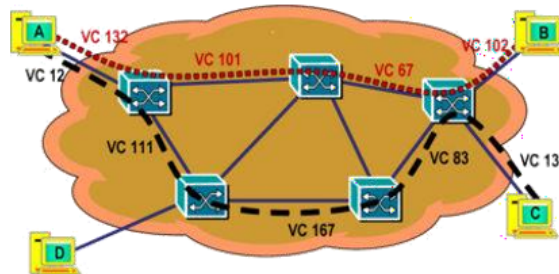
Storitve interneta

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

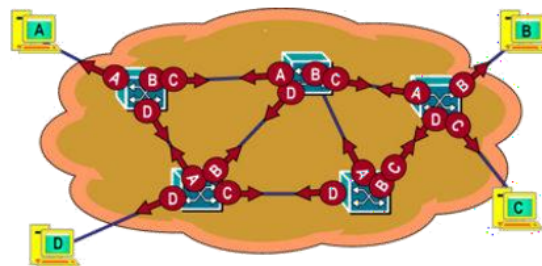
Best-effort service is a euphemism for no service at all.

Povezavna in nepovezavna omrežja

- **povezavna omrežja** (navidezni vodi) omogočajo vzpostavitev zveze v omrežni infrastrukturi med pošiljateljem in prejemnikom.

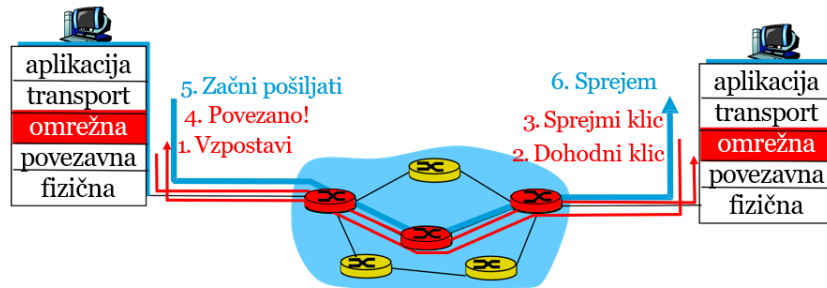


- **nepovezavna omrežja** (datagramska, paketna) omogočajo posredovanje paketov skozi infrastrukturo brez vzpostavljene povezave. Sem sodi tudi Internet.



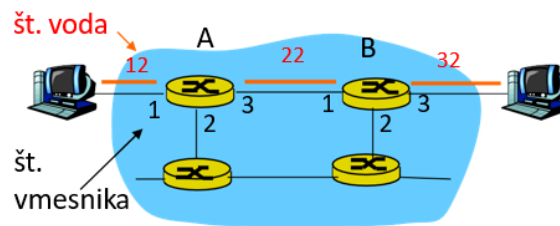
Navidezni vodi

- 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!)



Navidezni vodi: posredovalne tabele

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

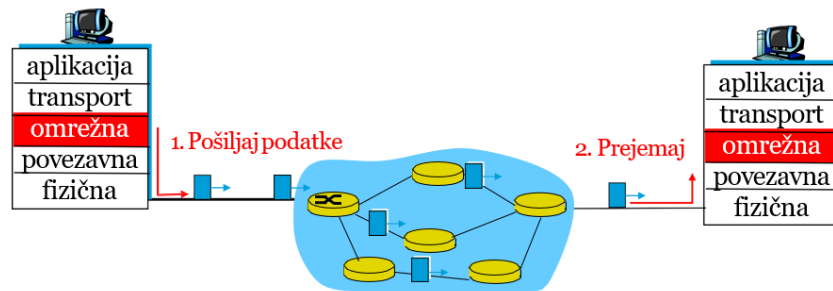


Vhodni vmesnik	Vhodna št. voda	Izhodni vmesnik	Izhodna št. voda
1	12	3	22
2	63	1	18
3	7	2	17
...

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

Usmerjevalniki morajo hraniti posredovalne tabele in stanje o povezavah. Za njihovo avtomatsko posodabljanje skrbijo **usmerjevalni algoritmi** - njihovo delovanje je počasno (v intervalu nekaj sekund - v primerjavi z vzpostavitvenim časom navideznih vodoov, ki je nekaj mikrosekund).



Datagramska omrežja: posredovalne tabele

- uporabljamo 32-bitne naslove pošiljateljev in prejemnikov
- Zaradi števila različnih naslovov, ne moremo hraniti tabele vseh naslovov. Možni rešitvi:
 - **Rešitev 1:** združimo dele naslovov v range (razpone naslovov) in vsakemu dodelimo vmesnik

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** (pregiksa) - začetnih bitov naslova. Posredujemo na vmesnik katerega ciljni naslov se najbolj ujema s prefiksom (torej če ustreza več predpon, izberemo najdaljšo) - ujemanje najdaljše predpone oz. longest prefix match

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

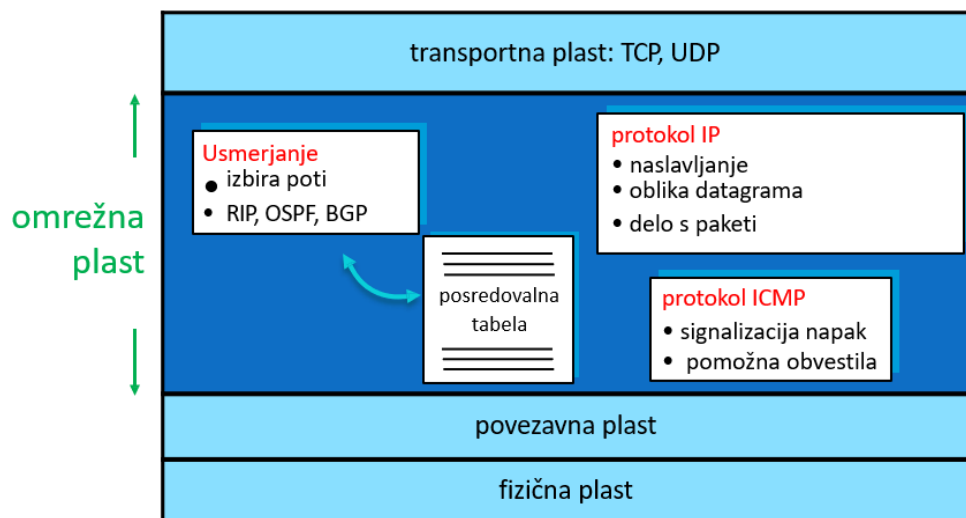
Primerjava obeh tipov omrežij

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

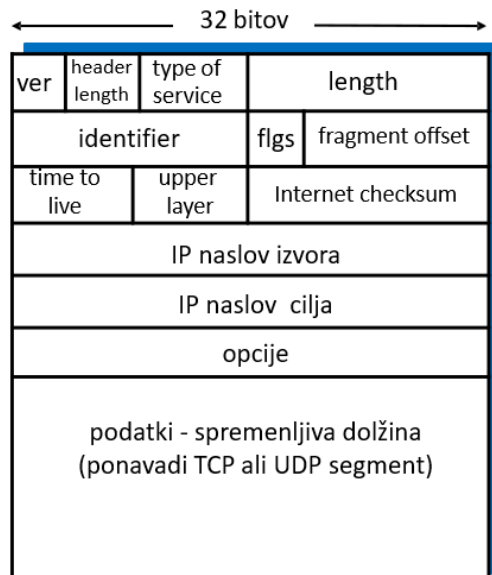
Internet Protocol(IP)

IP je ozko grlo interneta, ker deluje po modelu best-effort.

Funkcije omrežne plasti



IP: format paketa IPv4



- **VER**(4b): verzija protokola IP
- **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 1500B)
- **ID, FLAGS, OFFSET**(32b): potrebno za IP fragmentacijo
- **TTL**(8b): time to live, za preprečitev ciklanja datagramov po omrežju, vsak usmerjevalnik ga zmanjša za 1, tisti, ki ga zmanjša na 0 ga zavrže
- **UPPER LAYER PROTOCOL**(8b): polje za multipleksiranje - številka enkapsuliranega protokola v podatkih (6-TCP, 17-UDP)
- **CHECKSUM**(16b): kontrolna vsota (samo) glave datagrama, preračuna jo vsak usmerjevalnik - EDC biti
- **IP NASLOVI**(32b + 32b): naslovi izvora in cilja
- **OPCIJE**(32b): za možne razširitve glave datagrama(slabosti: večji čas potovanja, neznana lokacija začetka podatkov, običajno jih ni in je glava dolga 20B)
- **PODATKI**(spremenljiva dolžina), npr. ICMP se stlači v podatke

Fragmentacija

IP datagrami se morajo enkapsulirati v okvirje povezavne plasti, ti pa imajo omejeno dolžino(MTU - Maximum Transmission Unit, Ethernet do 1500B, 802.11:7918B). Fragmentacija razbije prevelik paket IP na več manjših (vsi so protokola IP, rečemo jim fragmenti).

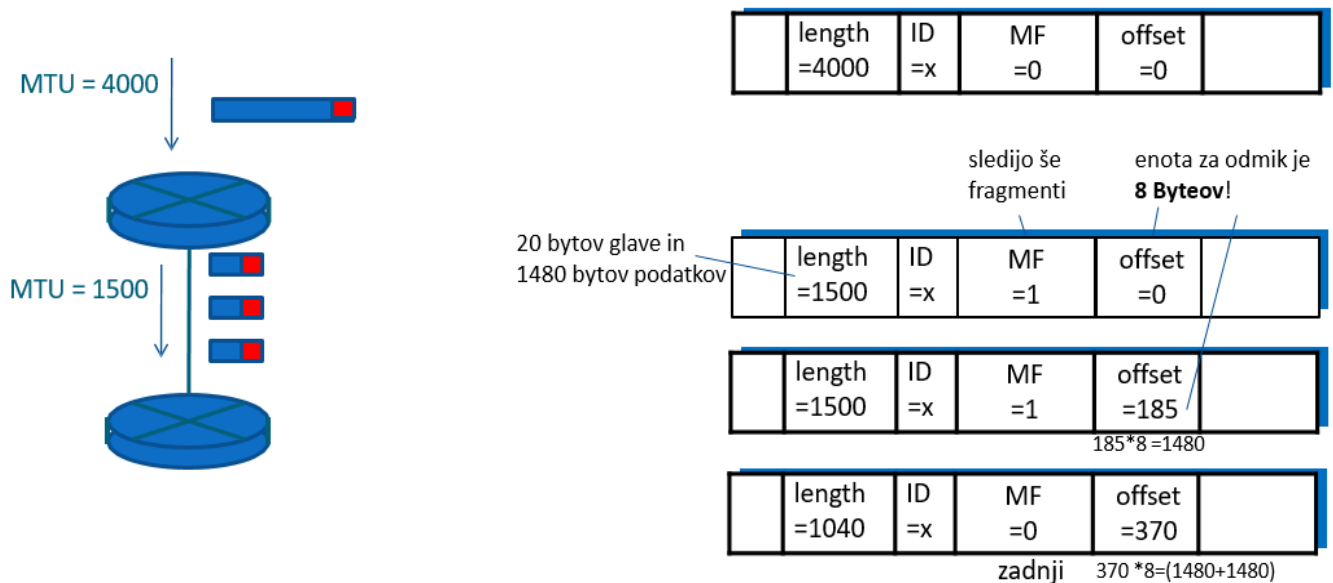
Pri tem se uporabljajo polja za fragmentacijo:

- **ID**: številka paketa(vrstni red), vsi fragmenti, ki pripadajo istemu paketu imajo enak ID
- **FLAGS**:
 - DF(don't fragment): če je 1 ne fragmentiraj
 - MF(more fragments): ta fragment še ni zadnji
- **OFFSET**: pozicija(odmik) fragmenta v prvotnem datagramu

V omrežju se lahko MTU med potjo spreminja, fragmentacijo lahko izvede tudi usmerjevalnik sredi poti. Fragmente združi šele omrežna plast prejemnika(torej usmerjevalnik jih posreduje tudi če imajo manjši MTU od maksimalnega) - usmerjevalniki po potrebi opravljajo le fragmentacijo. Pri IPv6 se prevelik paket zavrže, pošiljatelju se pošlje nazaj "Packet too big" in se mora pošiljatelj ukvarjati z razrešitvijo napake.

Primer fragmentacije

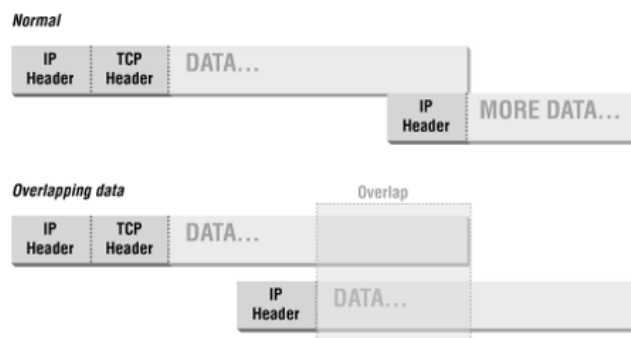
Imamo paket dolžine 4000B(20B glava + 3980B podatki), MTU pa je 1500B(20B glava + 1480B podatki).



Velikosti podatkov v fragmentih bodo 1480B + 1480B + 1020B.

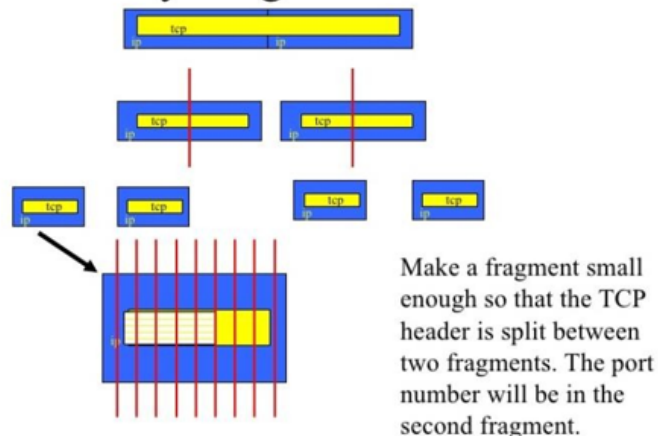
Napadi z uporabo fragmentacije

- **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)



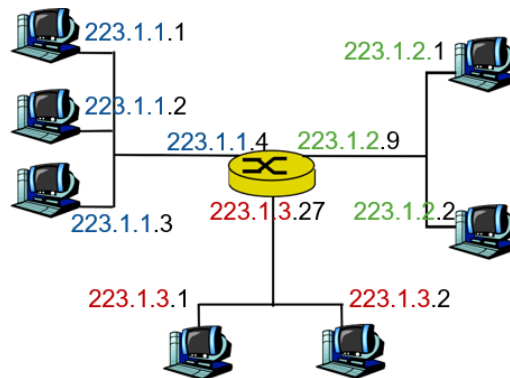
- **tiny fragment attack:** s fragmentacijo napadalec podatke razkosa tako, da fragmentira tudi podatke v glavi enkapsuliranega protokola. Na ta način ni možno izvesti varnostnega filtriranja po podatkih v glavi.

Tiny Fragment Attack



Podomrežje

- Usmerjevalnik ima na vsakem vmesniku("strani") drugo 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.



- **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.: "/25" za maskoS 11111111 11111111 11111111 10000000 ali 255.255.255.128



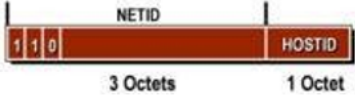
IPv4 naslavljanje

Vmesniki imajo IP(v4) naslove dožine 32 bitov. Računalniki imajo običajno en vmesnik, usmerjevalniki več. Naslovi naprav na Internetu morajo biti globalno unikatni. Spodaj je primer CIDR(Classless Inter-Domain Routing) naslavljanja.

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

Delitev na podomrežja

Sprva definirajo razrede(class) omrežij, ki uporabljajo maske iz 8, 16 ali 24 bitov. Kasneje se vpelje **prefiksna ali CIDR notacija**, ki omogoča dodelitev poljubnega števila bitov maski. Poseben IP naslov je *broadcast* naslov, ki naslovi vse naprave na podomrežju(naslov anprave so same enice)

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

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

Kako določimo IP naslove

- **Naprava:**

- administrator vpiše fiksen ali
- strežnik DHCP(Dynamic Host Configuration Protocol) 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 00010000 00000000	200.23.16.0/23
Podjetje2:	11001000 00010111 00010010 00000000	200.23.18.0/23
...

- **ISP:**

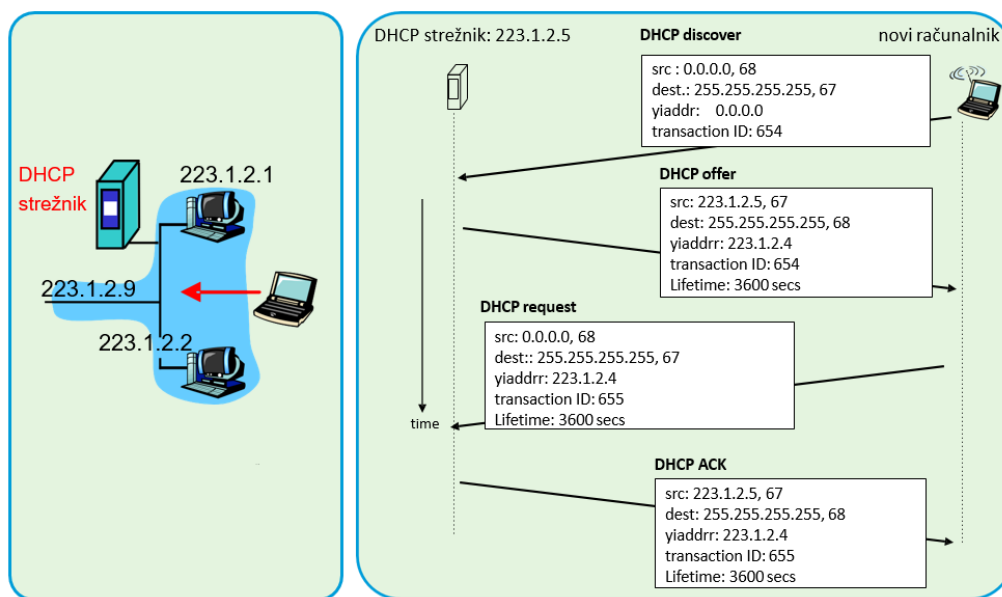
- Dodeli mu ga ICANN(Internet Corporation for Assigned Names and Numbers), neprofitna namenska organizacija, ki vzdržuje tudi korenske DNS strežnike

Dinamično dodeljevanje interneta

Ob priklopu naprava nima naslova IP, potrebna dodelitev osnovnih omrežnih nastavitev. DHCP(deluje na 7. plasti, vrata 67 ter 68) strežnik dodeli naslov v 4 fazah:

- DISCOVER
- OFFER
- REQUEST
- ACK

REQUEST in ACK sta tudi potrebni fazi zato, ker je lahko več DHCP strežnikov in tako končni sistem dobi več ponudb. Z ACK se odzove na tisto, ki jo sprejme.



NAT (Netowrk Address Translation)

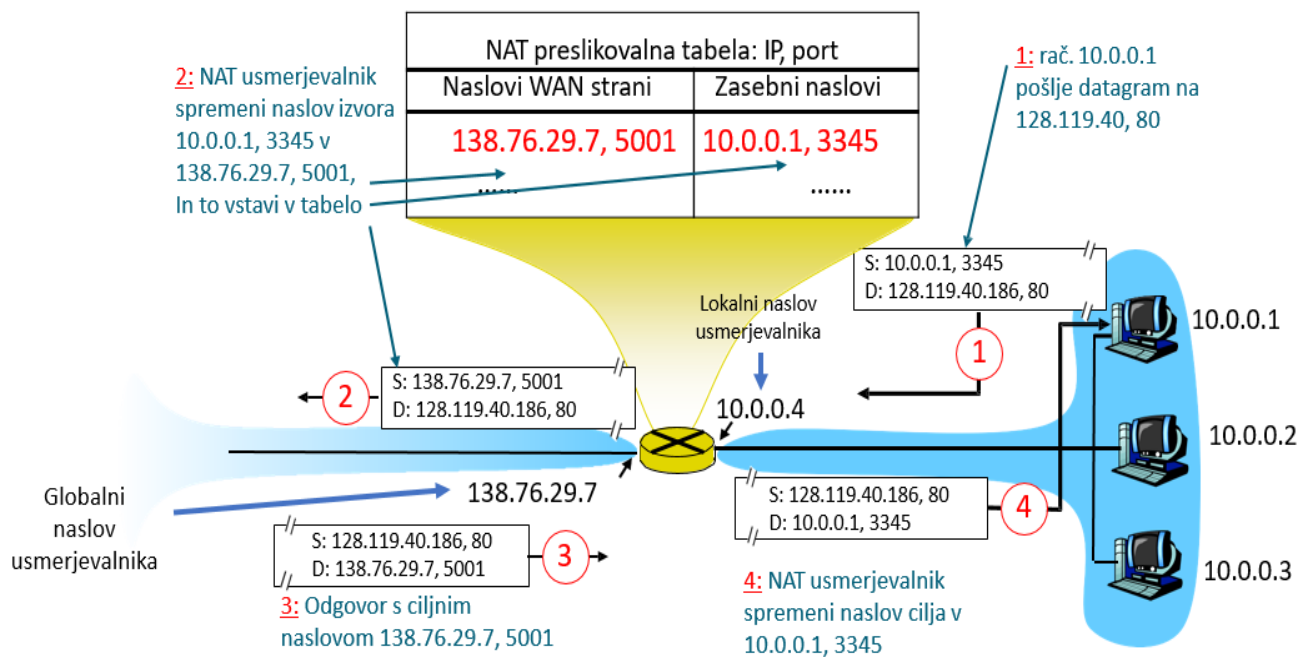
NAT: preslikovanje IP naslovov(pre-naslavljanje) se uvede zaradi pomanjkanja IPv4 naslovnega prostora.

Namesto, da trošimo unikatne javne(**globalne**) naslove, uporabljajmo raje **lokalne naslove**, ki so lahko **ponovljivi** med različnimi podjetji(ne nastopajo v javnem internetu). Usmerjevalnik uporablja NAT, da lokalni naslov preslika v globalni.

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}

Primer delovanja NAT



IP definira celoten končni sistem, vrata pe definirajo proces v končnem sistemu. Pri NAT-u pa dobijo številke vrat dodaten pomen, saj se jih uporabi za naslavljanje različnih končnih sistemov(od zunaj se vse pošilja na usmerjevalnikov IP, ta pa na podlagi vrat in vnosa v NAT tabeli posreduje ustreznemu končnemu sistemu).

Prednosti in slabosti 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!**

ICMP(Internet Control Message Protocol)

Se ošilja kot sporočilo, **enkapsulirano znotraj paketa IP(v podatkih)**(enako kot protokoli na transportni plasti). Uporablja se za imenjavo 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

Npr. PING uporablja tipa 0 in 8, traceroute pa izkorišča TTL(tip 11).

Traceroute: aplikacija, ki uporablja ICMP

Analizira po kateri poti gre promet do določenega cilja. Gre za koračni protokol, preizkuša komunikacijo na posameznih korakih(hop-ih) do cilja, da poišče pot skozi imrežje.

Delovanje 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 ni enak 0, ga zmanjša za 1 in posreduje naprej
 - Če je TTL enak 0, ga zavrže in pošle 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

Ni nujno, da se nekdo odzove na ICMP pakete. Če hočemo obdržati takšne informacije skrite, pač paket le posredujemo ali pa ga zavržemo, nanj se pa v nobenem primeru ne odzivamo.

Internet Protocol v6(IPv6)

Zakaj in kako?

- potreben je **večji naslovni prostor** (IPv4 naslovi so že izčrpani)
 - IPv6 omogoči naslove dolžine 128 bitov
 - tipičen(unicast) naslov je 64 bitov ID podomrežja + 64 bitov za ID vmesnika
- potrebno je **hitrejšo usmerjanje**
 - fiksna dolžina glave(40B), saj nimamo opcij
 - fragmentacija ni dovoljena, saj upočasnjuje procesiranje
- potrebno je zagotavljanje **kakovosti storitev**(QoS) za posebne tokove podatkov
 - oznaka "vrste toka"(flow label) v paketu IPv6

Sintaksa IPv6 naslova

IPv6 naslov v binarni obliki razdeljen na osem 16-bitnih skupin:

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

zapisan šestnajstičsko, 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**

□ **FF02:0:0:0:0:0:2** ali krajše **FF02::2**

□ **FF02:30:0:0:0:0:5** ni isto kot **FF02:3::5** (lahko pa zapišemo **FF02:30::5**)

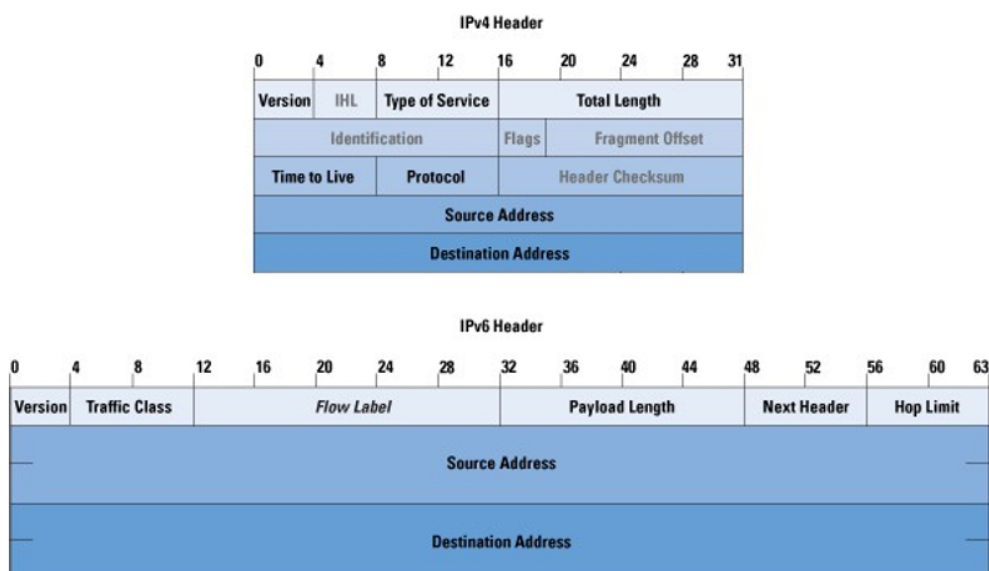
kompatibilnost z v4 naslovi: spredaj dodamo ničle

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

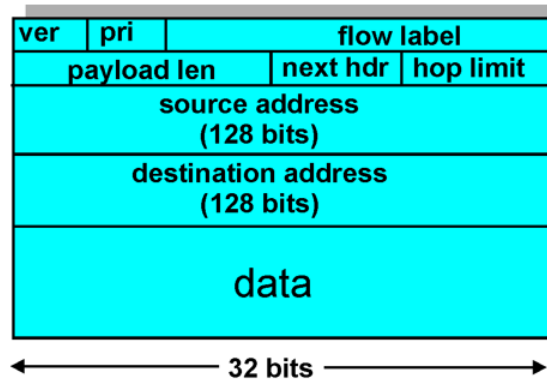
IPv6: hitrejša procesiranje paketov

- **fragmentacija se ne izvaja več** (usmerjevalnik imajo manj dela) - za delitev paketov skrbita pošiljatelj in sprejemnik. Če je paket prevelik ga usmerjevalnik zavrže in sporoči pošiljatelju "Packet Too Big"
 - nova verzija ICMP - ICMPv6(eno sporočil tega protokola je tudi "Packet Too Big")
- glava več **ne vsebuje kontrolne vsote**, saj je ta prisotna že v enkapsuliranih protokolih znotraj IP in zavira procesiranje(ko usmerjevalnik spremeni TTL mora ponovno preračunati tudi kontrolno vsoto)
- **polja za opcije v glavi ni več**. Možno jih je implementirati na 2 načina:
 - lahko konstruiramo lasten protokol
 - lahko konstruiramo obliko lastnega aplikacijskega sporočila(veliko lažje kot konstruirati svoj protokol) in ga enkapsuliramo v podatke

Primerjava paketov IPv4 in IPv6



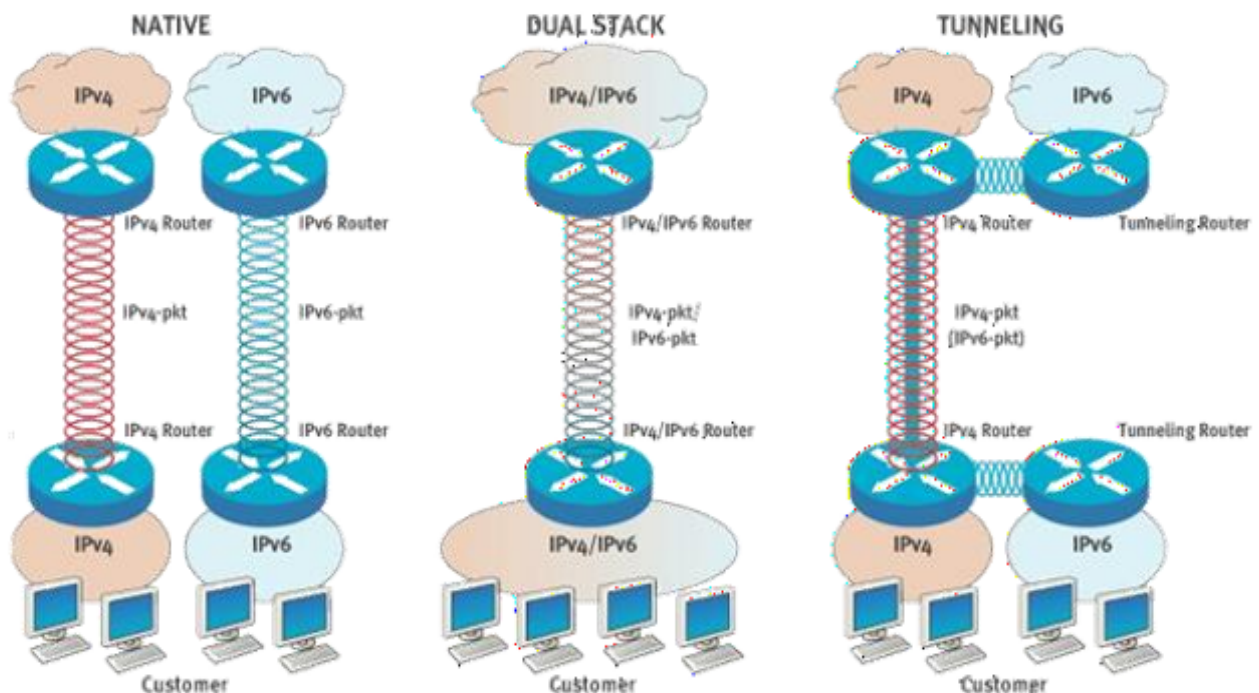
IPv6 paket



- **VER**(4b): verzija protokola IP
- **PRI ali TRAFFIC CLASS**(8b): oznaka prioritete, podobno kot Type of Service pri IPv4
- **FLOW LABEL**(20b): oznaka "toka" podatkov, ki omogoča 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

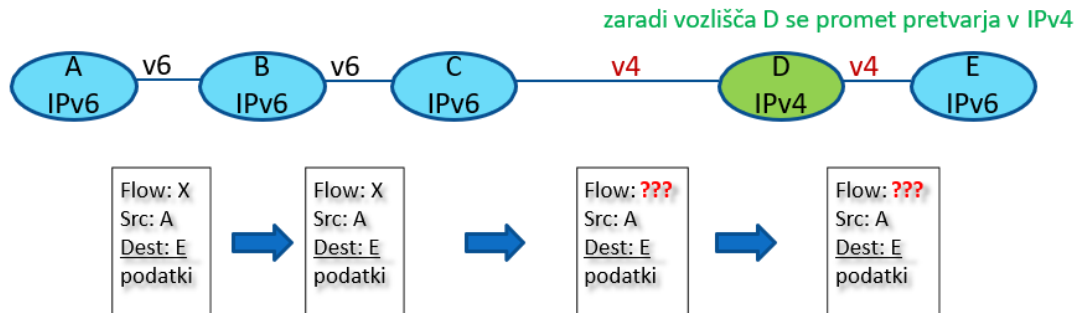
Prehod iz IPv4 na IPv6

- vseh naprav ni mogoče nadgraditi naenkrat
- za prehod se najpogosteje uporabljata dve rešitvi:
 - dvojni sklad(dual-stack, vozlišča uporabljajo vzporedni implementaciji IPv4 in Ipv6), ko se da, se uporabi IPv6
 - tuneliranje(tunneling, paket IPv6 zapakiramo v paket IPv4 kot podatke)

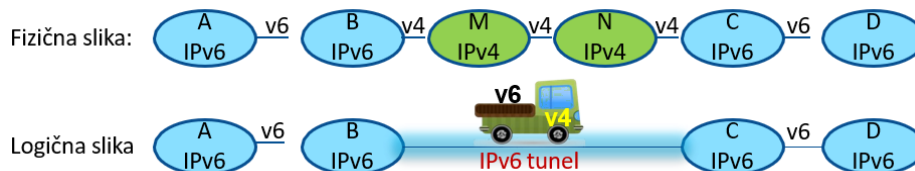


Dvojni sklad

- Usmerjevalnik "govori" IPv4 in IPv6. IPv4 uporablja samo, ko ne more IPv6
- Naprava ugotovi katero različico uporabljati pri povpraševanju za IP glede na to ali ji DNS strežnik vrne IPv4 ali IPv6 naslov
- Če je na poti med dvema IPv6 vozliščema kakšno IPv4 se bo promet pretvarjal v IPv4, pri tem pa se bodo izgubila specifična polja IPv6(flow label)



Tuneliranje



Naloga od B je le, da enkapsulira paket v IPv4. B ne ve, za koliko usmerjevalniki je C in ga tudi ne zanima. B mora konfigurirati administrator, da vzpostavi tunel, nato lahko komunikacija steče. Če se paket IPv4 v tunelu fragmentira, ga mora C najprej zložiti in nato dekapsulirati v IPv6.

Varnost na omrežni plasti: IPSec

Komunikacija na omrežni plasti poteka nevarovano(možna so ponarejanja naslovov, prisluškovanje ipd.). IPSec je nabor protokolov, ki skrbijo za varno komunikacijo na omrežnem nivoju(AH - Authentication Header, ESP - Encapsulating Security Payload).

Storitve:

- dogovor o uporabljenih kriptografskih algoritmi in ključih
- enkripcija in dekripcija
- integriteta podatkov
- avtentikacija izvora

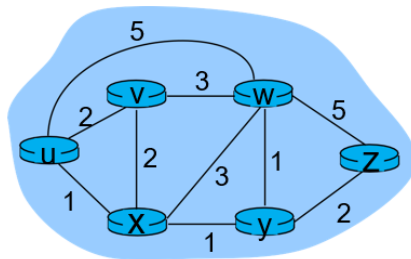
Usmerjanje

Omrežja modeliramo s teorijo grafov $G = \langle V, P \rangle$

- vozlišča V: usmerjevalniki
- povezave P: komunikacijske povezave

Usmerjevalni protokoli

Konfigurirajo posredovalne tabele v usmerjevalnikih, da s tem omogočijo vzpostavitev **najcenejše** poti. Cena je lahko razdalja, denar, hitrost... Ceno je potrebno natančno definirati pred usmerjanjem.



- $c(x, x') = \text{cena povezave } (x, x')$
- **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**

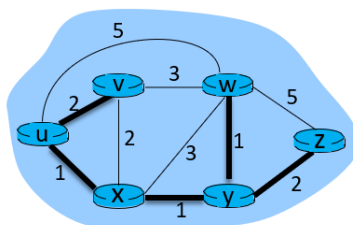
Vrste usmerjevalnih algoritmov

Možne so različne kombinacije lastnosti:

- **centralizirani**(globalni) ali **decentralizirani**(porazdeljeni):
 - centralizirani imajo dostopne podatke o stanju povezav v celem omrežju (link state algoritmi) - usmerjanje na podlagi stanja povezav v omrežju
 - decentralizirani imajo dostopne podatke samo o neposredno priključenih povezavah. Izračun optimalne poti poteka iterativno.
- **prilagodljivi** ali **neprilagodljivi** na obremenitev povezav:
 - prilagodljivi avtomatsko prilagajajo cene povezav glede na zasičenost povezave(bolj proste poti dobijo manjšo ceno)

Centralizirani(globalni) algoritmi

- možna uporaba **centralnega vozlišča**, ki koordinira usmerjanje ali neodvisno izračunavanje posameznih vozlišč
- vsako vozlišče sporoča stanje povezav vsem ostalim vozliščem(preko broadcast-a v IPv4 oz. multicast-a v IPv6)
- vsako vozlišče izračuna **drevo najkrajših poti do ostalih vozlišč**(algoritem Dijkstra) - rezultat je posredovalna tabela na to vozlišče

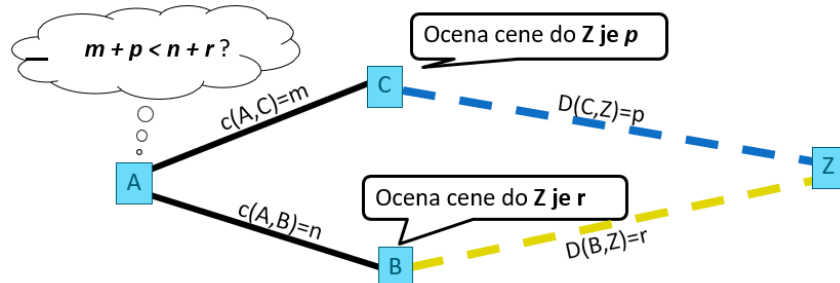


Izračunana posredovalna tabela vozlišča u

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

Porazdeljeno usmerjanje

- usmerjanje z **vektorjem razdalj**
- vsako vozlišče izračuna posredovalno tabelo na osnovi lokalnih podatkov, prejetih od sosedov
- usmerjanje je **iterativno** - računanje 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 sosedu s do cilja y : $D(s,y)$



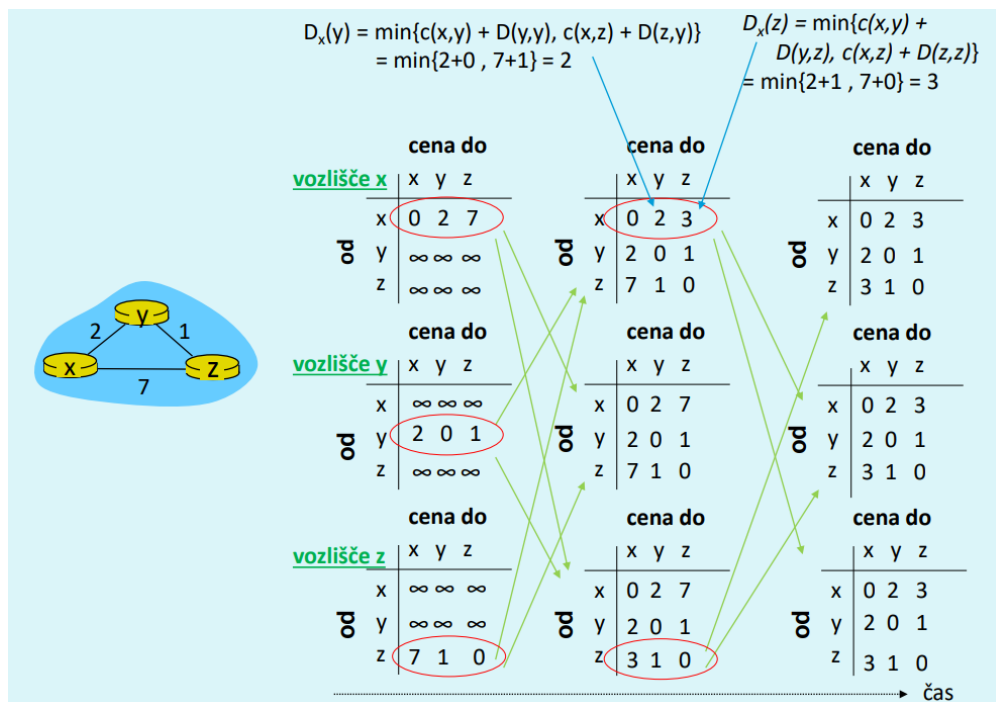
Usmerjanje z vektorjem razdalj

Vsako vozlišče hrani:

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



Osnovna ideja:

- Ko vozlišče x prejme vektor razdalj od sosedu s , izračuna svoj novi vektor razdalj: $D(x,y) \leftarrow \min\{c(x,s) + D(s,y)\}$ za vsa vozlišča y
- Vsako vozlišče pošle svoj vektor razdalj običajno takrat, ko zazna spremembo v svojem vektorju razdalj
- Sčasoma cene konvergirajo v dejanske najmanjše cene



Širjenje novic(sprememb cen) - porazdeljeno usmerjanje

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

odkrivanje nove poti do A					prekinitev povezave do A				
									
začetek	∞	∞	∞	∞	začetek	1	2	3	4
1. iteracija	1	∞	∞	∞	1. iteracija	3	2	3	4
2. iteracija	1	2	∞	∞	2. iteracija	3	4	3	4
3. iteracija	1	2	3	∞	3. iteracija	5	4	5	4
4. iteracija	1	2	3	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
 - administrator vsakega omrežja želi administrirati 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 vektor ali link-state)
 - 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 preusmeriti pakete v destinacije iz 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(vektor se osvežuje na 30s)
 - cena se optimizira na podlagi hopov (min = 1, max = 15)
 - **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), danes najbolj popularen protokol
 - omogoča, da omrežja oglašujejo svojo prisotnost drugim omrežjem