

# Računalniške komunikacije

2023/24

povezavna plast  
storitve povezavne plasti  
protokoli za dostop do medija  
Ethernet  
ARP  
stikala  
VLAN

# Pridobljeno znanje s prejšnjih predavanj

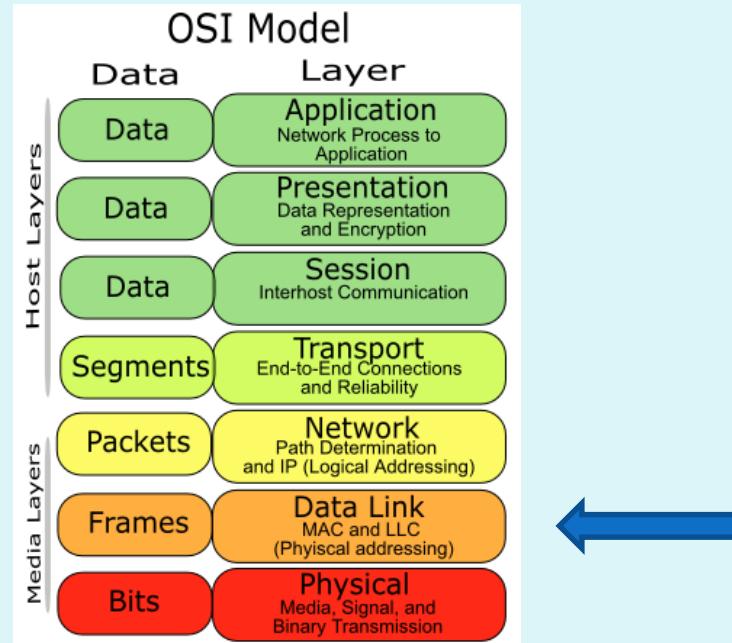
- definicija omrežij (fizična in storitvena)
- komponente omrežja (končni sistemi, jedro, povezave)
- načini priklopa v omrežje (modem, DSL, kabelski, optični, ozičeni-Ethernet, brezžični- 3G/4G/802.11)
- vrste omrežij glede na delovanje: povezavno, nepovezavno
- kaj je to komunikacijski protokol
- plasti, pari procesov na plasteh, odnosi med nadrejenimi/podrejenimi plastmi
- ISO/OSI in TCP/IP model ter njune plasti
- naloge fizične plasti
- vrste prenosnih medijev

# Kje smo?

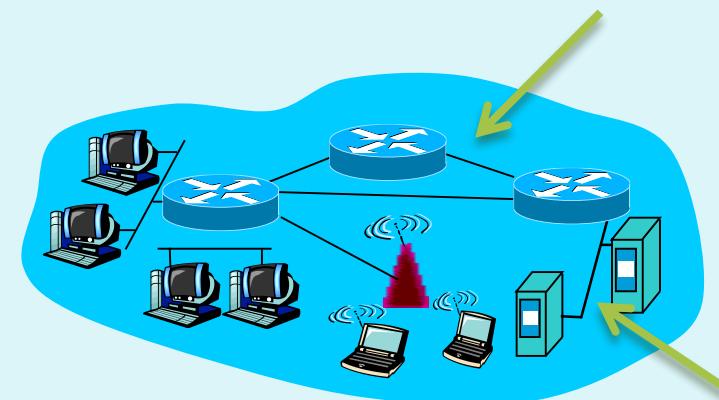
- **fizična plast:** plast, ki določa način kodiranja bitov v signal
  - modulacije valovanj: amplitudna, frekvenčna, fazna, kvadratna
- **prenosni medij:** naprava, ki omogoča razširjanje valovanja
  - tipi medijev: bakreni, optični, brezžični (radijski, IR, Bluetooth)



# Povezavna plast



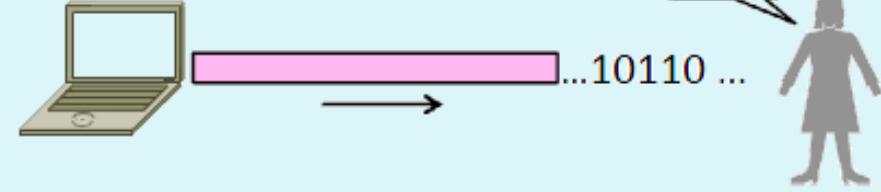
- enota, ki se prenáša na povezavni plasti je OKVIR (angl. *frame*)
- NALOGA povezavne plasti:  
prenos okvirja *po povezavi med sosednjima vozliščema* (računalnik, usmerjevalnik) upoštevajoč **tip medija**



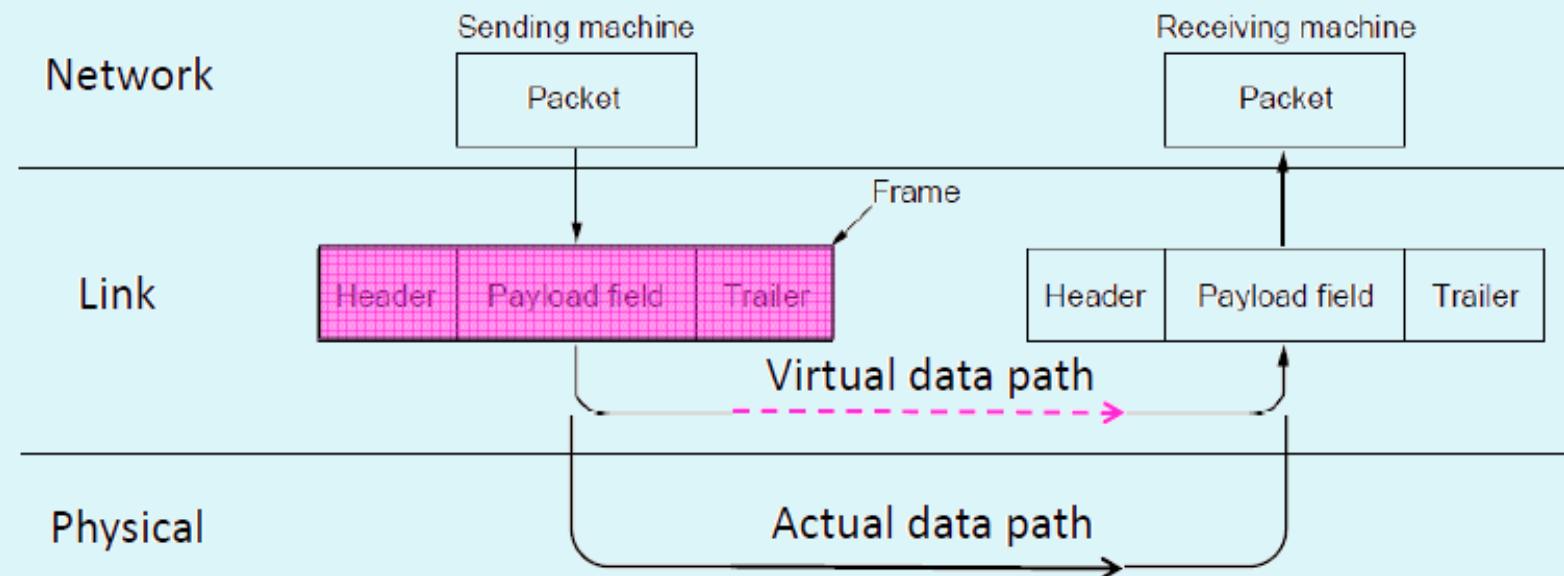
# Storitve povezavne plasti

- povezavna plast **lahko** izvaja:
  1. **okvirjanje datagramov**: podatkom višje plasti se doda glava in določi struktura
  2. **zaznavanje in odpravljanje napak**: z dodatnimi biti lahko zaznavamo, ali je prišlo do napake pri prenosu okvirja; v določenih primerih jo lahko odpravimo
  3. **dostop do medija**: če je medij deljen, se uporablja MAC protokol (*media access control*) in ustrezeno naslavljjanje udeležencev
  4. **zagotavljanje zanesljive dostave**: uporaba potrjevanja in ponovnega pošiljanja v primeru napake pri prenosu na povezavi
  5. **kontrola pretoka**: usklajevanje hitrosti pošiljanja glede na procesorske sposobnosti prejemnika

# Kaj je okvir?



- "enota" podatkov na povezavni plasti
- opredeljuje začetek in konec prenesenih podatkov
- podatkom doda glavo (*header*) in rep (*trailer*), ki so potrebni za uspešen prenos



# Protokoli na povezavni plasti

- komunikacija je lahko sestavljena iz različnih omrežnih povezav in medijev -> uporaba **različnih protokolov!**
- vsak protokol definira **svojo obliko** izmenjanih podatkov med vozlišči
- **primeri** protokolov:
  - Ethernet
  - wireless LAN (802.11)
  - token ring
  - PPP

# Analogija iz realnega sveta

- cilj potovanja: pot iz Loma na morje v Koper



potnik je "podatkovni paket"

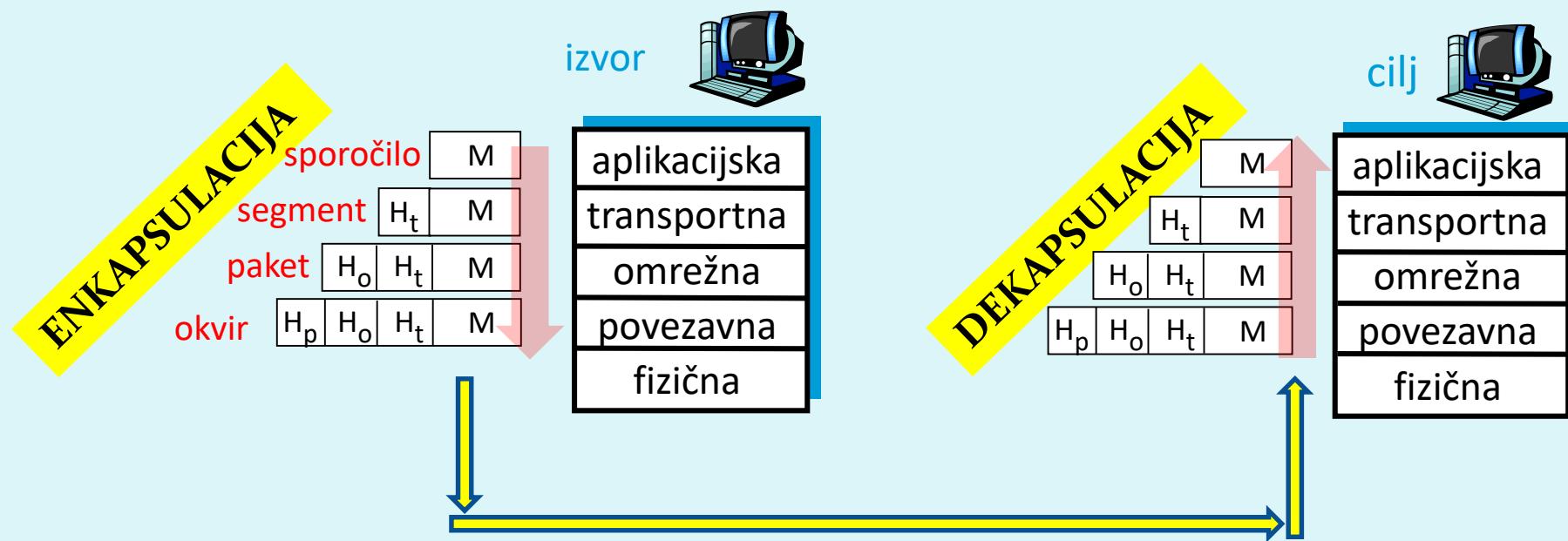
del poti je "komunikacijska povezava"

začetek in cilj sta "končni vozlišči"

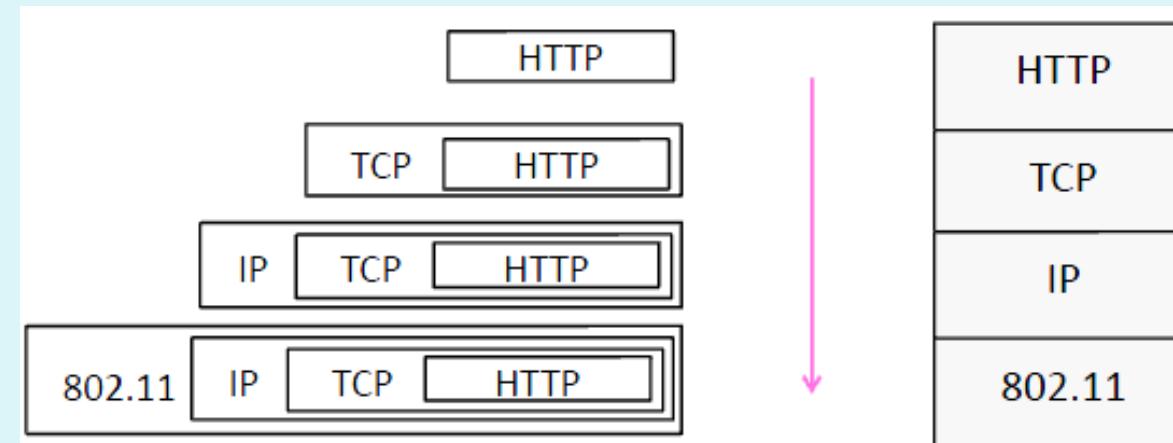
način prevoza je "protokol povezavne plasti"

naš organizator potovanja od začetka do cilja je "usmerjevalni protokol" omrežne (3. plasti!)

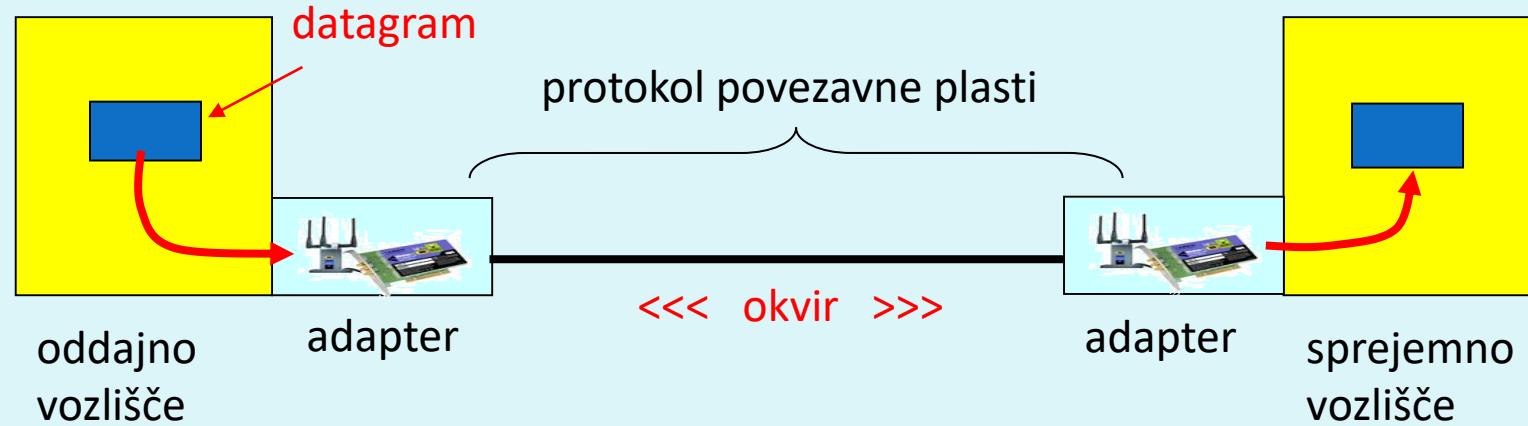
# Enkapsulacija in dekapsulacija



# Enkapsulacija: primer



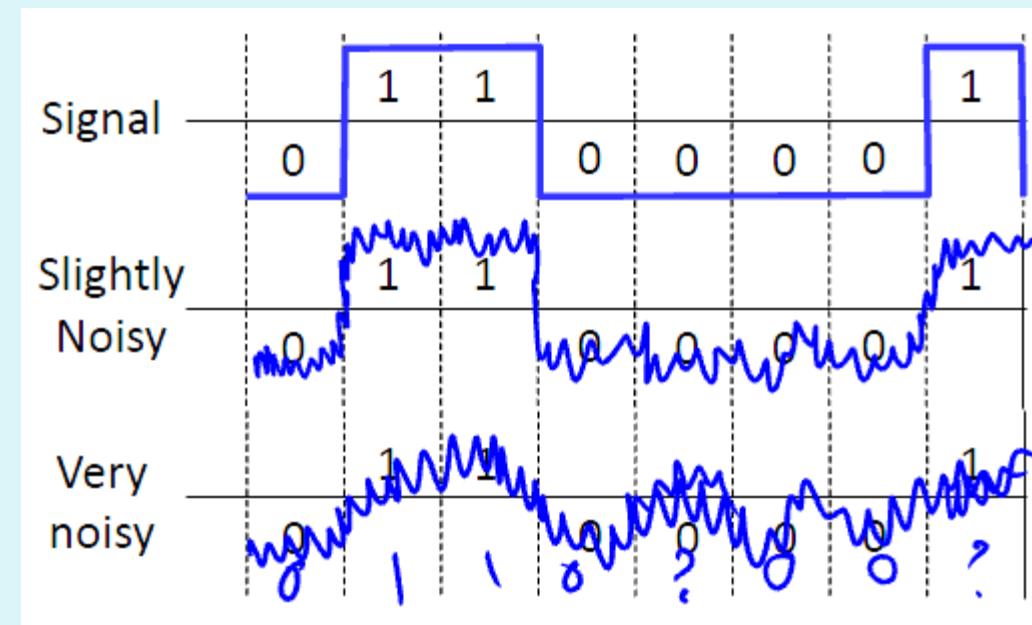
# Implementacija povezavne plasti



- **oddajnik:** enkapsulacija datagrama v okvir, detekcija, kontrola pretoka ...
  - enkapsulacija: podatki se zapakirajo v okvir, izpolnijo se dodatna kontrolna polja (naslovi pošiljatelja, prejemnika itd.)
- **sprejemnik:** preveri napake, pretok, dekapsulacija.

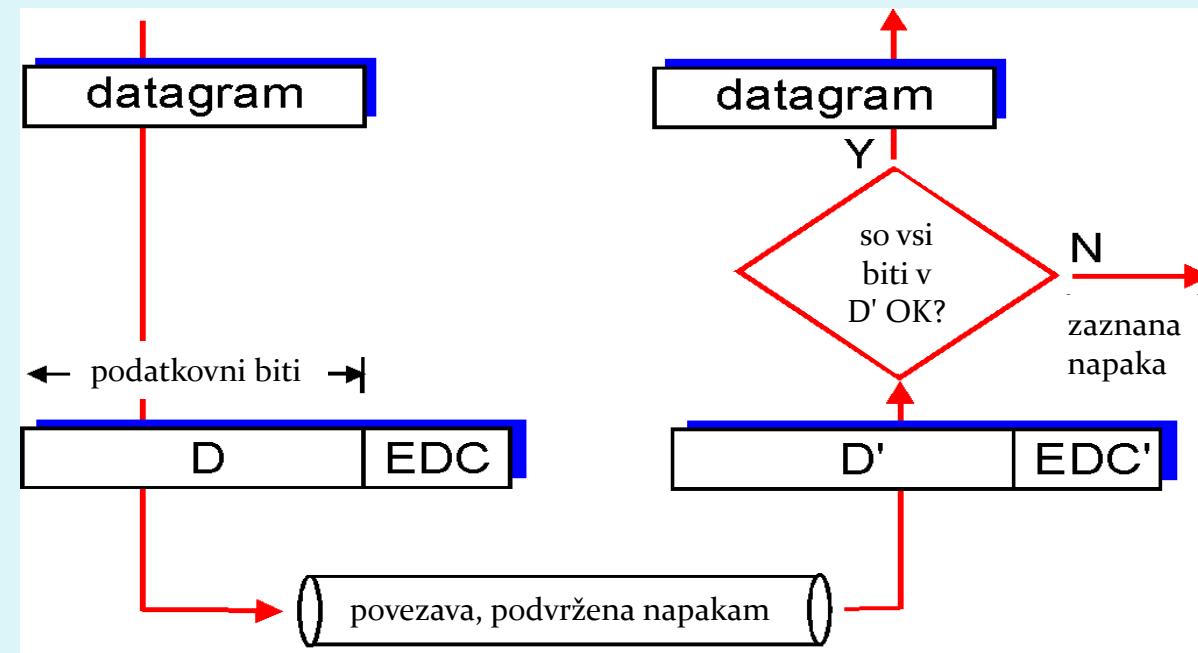
# Zaznavanje in odpravljanje napak (1/5)

- zakaj je potrebno?
  - motnje na kanalu: presluh, slabljenje, šum
  - signal se lahko okvari, da prejemnik zamenja vrednosti 0 in 1



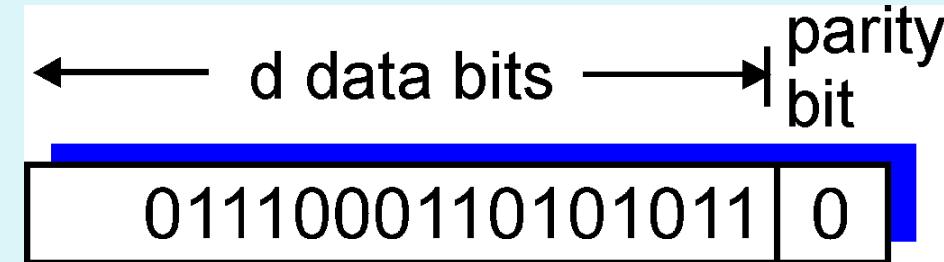
# Zaznavanje in odpravljanje napak (2/5)

- Podatkom (D) dodamo še dodatne bite za preverjanje pravilnosti (EDC - *Error Detection Code*)
  - protokol za popravljanje ni popoln, lahko spregleda napake,
  - več EDC bitov omogoča boljšo detekcijo/popravljanje



# Zaznavanje in odpravljanje napak (3/5)

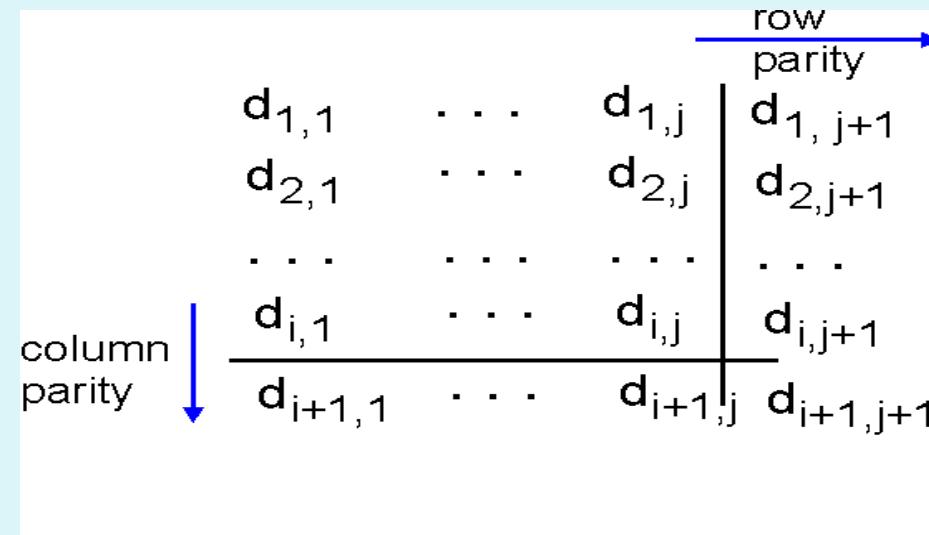
- parnost: dodamo 1 paritetni bit
  - soda/liha paritetna shema
  - primer: liha paritetna shema (pariteta=0, če imamo v podatkih liho število enic in 1, če imamo sodo število enic)



- uspešnost?
  - omogoča samo zaznavanje lihega števila napak

# Zaznavanje in odpravljanje napak (4/5)

- parnost v 2 dimezijah: dodamo paritetne bite za vsako vrstico in stolpec
  - omogoča zaznavanje in odpravljanje enojnih ali celo dvojnih napak
  - (možno tudi zaznavanje napak v paritetnih bitih)



101010  
111101  
011100  
110101  
*no errors*

101010  
101100  
011100  
100101  
parity error  
parity error  
*correctable single bit error*

# Popravljanje s Hammingovo kodo

- Primer: podatek=0101, dodamo 3 kontrolne bite
  - izračun kontrolnih bitov s sodo paritetno shemo

- 7 bit code, check bit positions 1, 2, 4
- Check 1 covers positions 1, 3, 5, 7
- Check 2 covers positions 2, 3, 6, 7
- Check 4 covers positions 4, 5, 6, 7

0 1 0 0 1 0 1 →  
1 2 3 4 5 6 7

$$p_1 = 0+1+1 = 0, \quad p_2 = 0+0+1 = 1, \quad p_4 = 1+0+1 = 0$$

pravilen prenos

→ 0 1 0 0 1 0 1  
1 2 3 4 5 6 7

$$p_1 = 0+0+1+1 = 0, \quad p_2 = 1+0+0+1 = 0, \\ p_4 = 0+1+0+1 = 0$$

Syndrome = 000, no error

Data = 0 1 0 1

korekcija napake

→ 0 1 0 0 1 1 1  
1 2 3 4 5 6 7

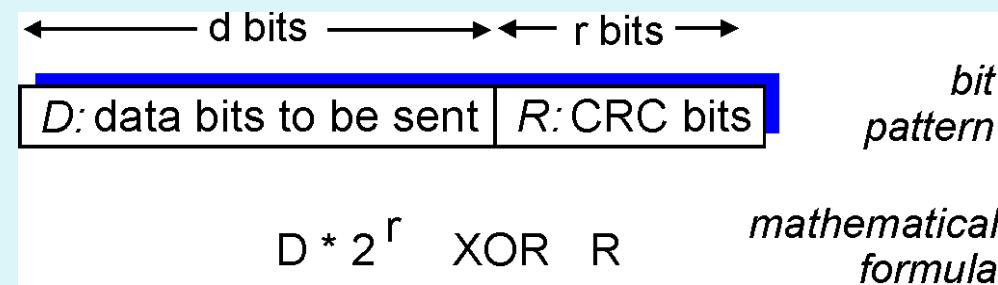
$$p_1 = 0+0+1+1 = 0, \quad p_2 = 1+0+\textcolor{red}{1}+1 = \textcolor{red}{1}, \\ p_4 = 0+1+\textcolor{red}{1}+1 = \textcolor{red}{1}$$

Syndrome = **1 1 0**, flip position 6

Data = 0 1 0 1 (correct after flip!)

# Zaznavanje in odpravljanje napak (5/5)

- uporaba kontrolnih vsot:
  - CRC (angl. *Cyclic Redundancy Check*): matematična metoda, ki uporablja polinome. Uporablja r dodatnih bitov, sposobna zaznati in popraviti napake do r+1 bitov



- kasneje (transportna plast) bomo spoznali še eno kontrolno vsoto - Internetna kontrolna vsota

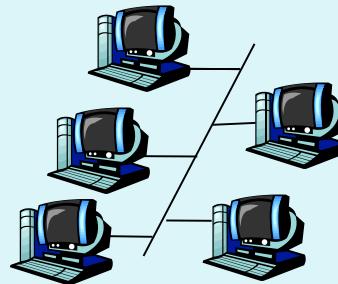
# Storitve povezavne plasti

- povezavna plast izvaja:
  1. **okvirjanje datagramov:** podatkom višje plasti se doda glava in določi struktura
  2. **zaznavanje in odpravljanje napak:** z dodatnimi biti lahko zaznavamo, ali je prišlo do napake pri prenosu okvirja; v določenih primerih jo lahko odpravimo
  3. **dostop do medija:** če je medij deljen, se uporablja MAC protokol (*media access control*) in ustrezeno naslavljjanje udeležencev
  4. **zagotavljanje zanesljive dostave:** uporaba potrjevanja in ponovnega pošiljanja v primeru napake pri prenosu na povezavi
  4. **kontrola pretoka:** usklajevanje hitrosti pošiljanja glede na procesorske sposobnosti prejemnika



# Protokoli za dostop do skupinskega medija

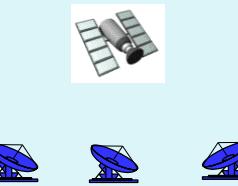
- dve vrsti povezav:
  - *dvotočkovna (point-to-point)* povezava: vsaka povezava ima le enega pošiljatelja in prejemnika (npr. protokola PPP, HDLC)
  - *oddajna (broadcast)* povezava: deljeni medij, več vozlišč komunicira naenkrat (npr. Ethernet, Wireless LAN)



deljeni kabel



deljen frekv. spekter  
(e.g., 802.11 WiFi)



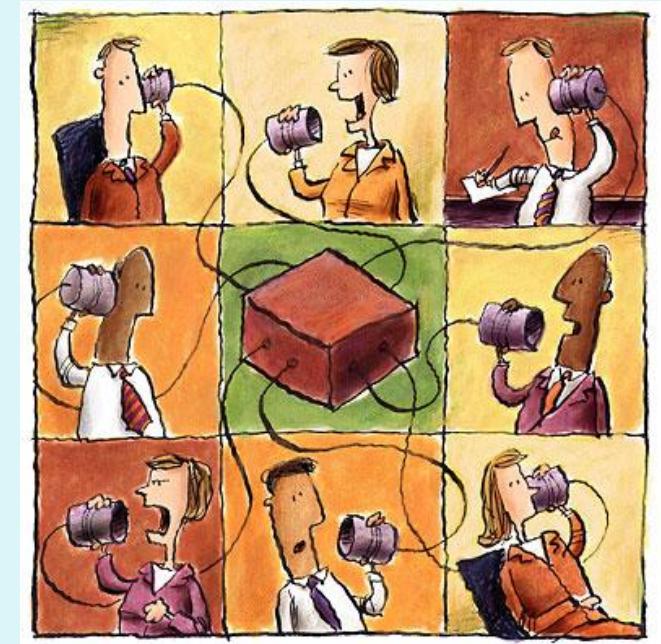
deljen satelitski  
signal



analogija: pogovor na  
zabavi (medij: zrak,  
ki prenaša govor)

# Protokoli za dostop do skupinskega medija

- oddajni (broadcast) kanali:
  - potreben protokol za koordinacijo dostopa (kdo lahko govori), t. i. *multiple access*
- principi iz realnega sveta, ki jih upoštevamo:
  - daj vsakemu priložnost, da govori
  - ne odgovarjaj, razen če te kdo ne ogovori
  - ne izvajaj monologov (ne monopoliziraj pogovora)
  - dvigni roko, če imaš vprašanje
  - ne prekinjaj nekoga, ko ta govori
  - ne spi, ko ti nekdo govori
- če dve vozlišči oddajata naenkrat, pride do TRKA ali KOLIZIJE (signal se preplete in okvari)



# Protokol za dostop do skupinskega medija

Za **idealni protokol**, ki upravlja dostop do kanala s hitrostjo  $R$  velja:

1. če oddaja eno samo vozlišče, oddaja s hitrostjo  $R$   
(izkoristek: eno vozlišče lahko izkoristi polno kapaciteto)
2. če oddaja  $M$  vozlišč, oddajajo s povprečno hitrostjo  $R/M$   
(pravičnost: vsak pride do svojega deleža kapacitete)
3. protokol je decentraliziran  
(kolektivnost: ni centralnega vozlišča, ki ga upravlja)
4. je enostaven

# Izogibanje in razreševanje kolizij

## 1. delitev kanala (ni kolizij):

kanal razdelimo na "podkanale" (frekvečno ali časovno) in vsakega dodelimo paru vozlišč

## 2. naključni dostop (kolizije so):

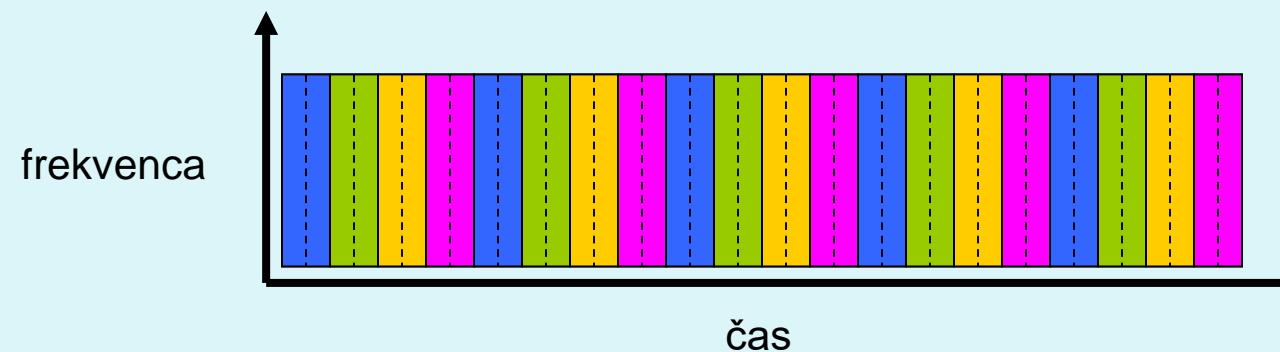
vsak lahko oddaja kadarkoli, če pride do kolizije, jo razrešujemo

## 3. izmenični dostop (ni kolizij):

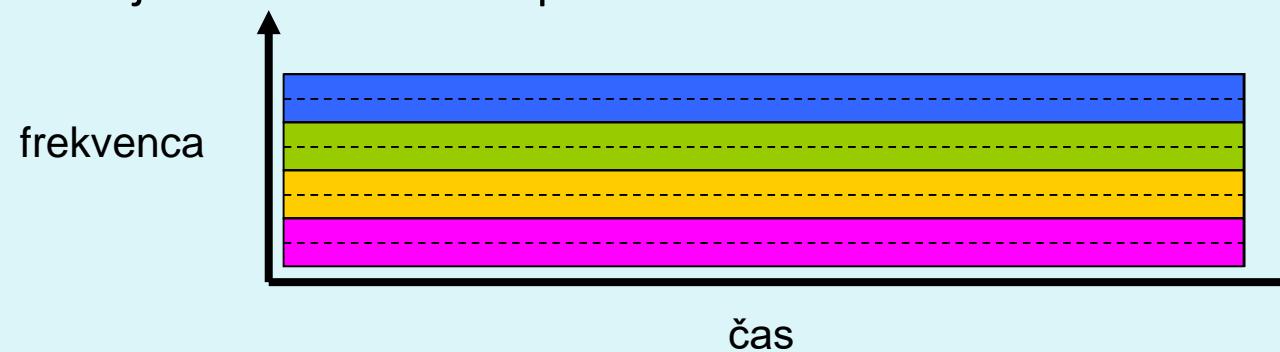
vozliščem izmenično dodelujemo pravico do pošiljanja

# 1. Delitev kanala

- TDMA (Time Division Multiple Access)
  - v vsakem “krogu” vsaka postaja dobi enak časovni interval (1 paket)



- FDMA: Frequency Division Multiple Access
  - vsaka postaja ima svoj fiksen frekvenčni pas



# 1. Delitev kanala

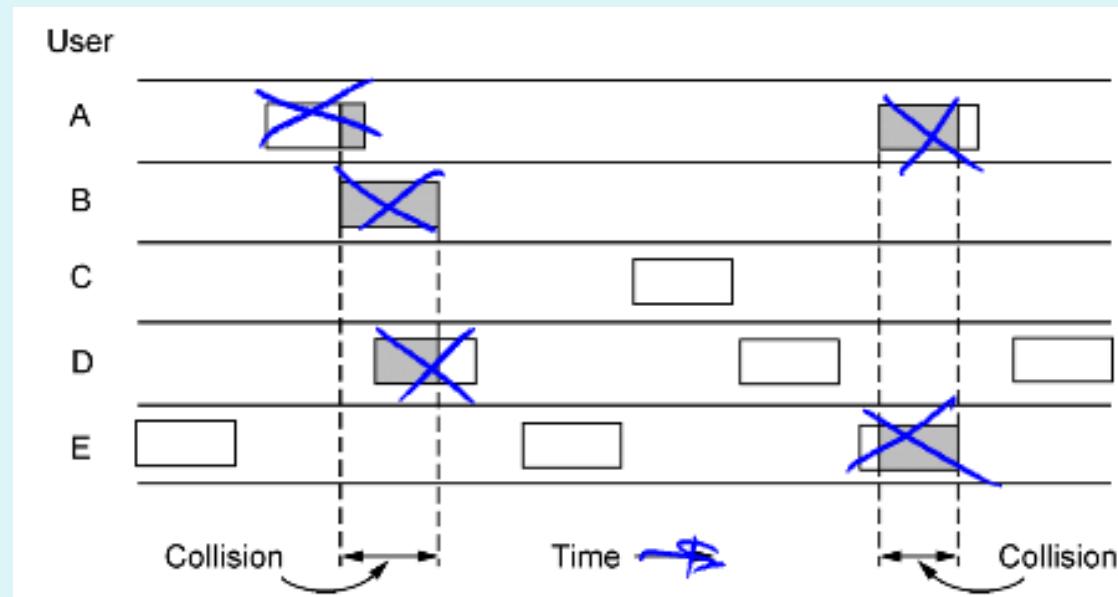
- pošteno in učinkovito pri visoki obremenitvi
- pri majhni obremenitvi se pojavlja neizkoriščenost kanala
- potrebno čakanje na vrsto
- upoštevanje smernic za *idealni protokol*:
  1. če oddaja eno samo vozlišče, oddaja s hitrostjo R 
  2. če oddaja M vozlišč, oddajajo s povprečno hitrostjo  $R/M$  

## 2. Protokoli za naključni dostop

- Določajo:
  - kako zaznati kolizijo
  - kako ukrepati ob koliziji
- Kadar želi vozlišče pošiljati, uporabi polno razpoložljivo hitrost kanala  $R$ . Pred pošiljanjem ni koordinacije med vozlišči.
- Primeri MAC protokolov z naključnim dostopom:
  - ALOHA
  - razsekana ALOHA
  - CSMA, CSMA/CD, CSMA/CA

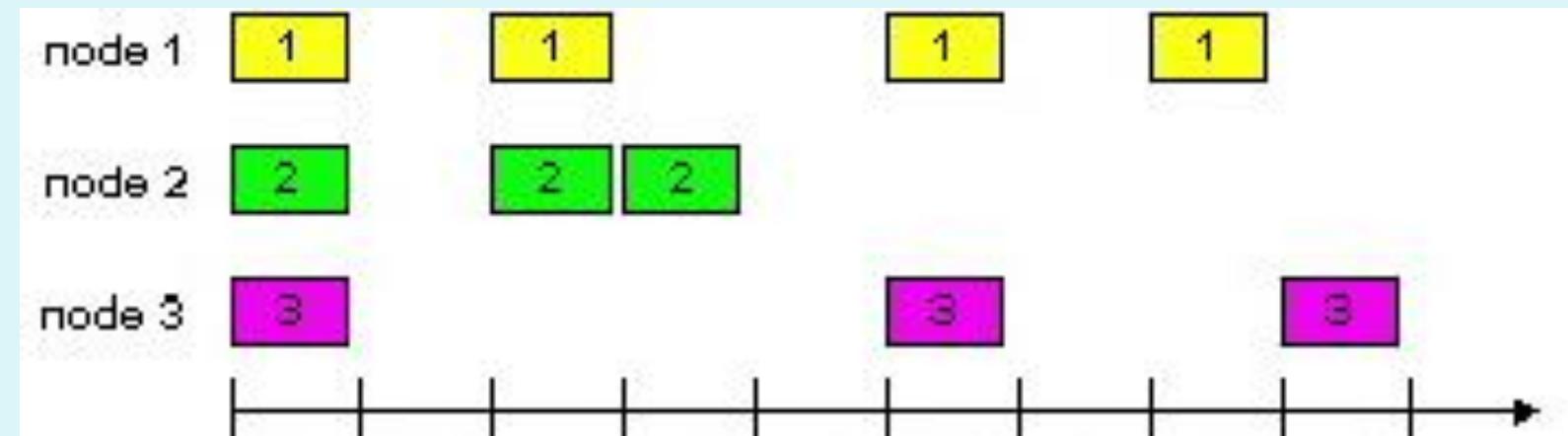
# ALOHA

- paket je ranljiv ves čas oddajanja (čas je zvezen)
- preprost, ni sinhronizacije
- obravnavanje kolizije:
  - paket se pošlje do konca (!) in ponovno pošlje po preteku naključnega intervala časa



# Razsekana ALOHA

- čas je razsekan na enake časovne intervale (v katerih je možno poslati 1 okvir)
- vozlišča so sinhronizirana, pošiljajo samo ob začetku intervalov,
- če pride do kolizije:
  - vozlišče pošlje ponovno okvir v naslednjem intervalu z verjetnostjo  $p$
- +: vozlišče lahko uporablja celo hitrost  $R$ , enostaven
- -: prazni/kolizijski intervali so neuporabni, potrebna je sinhronizacija časa

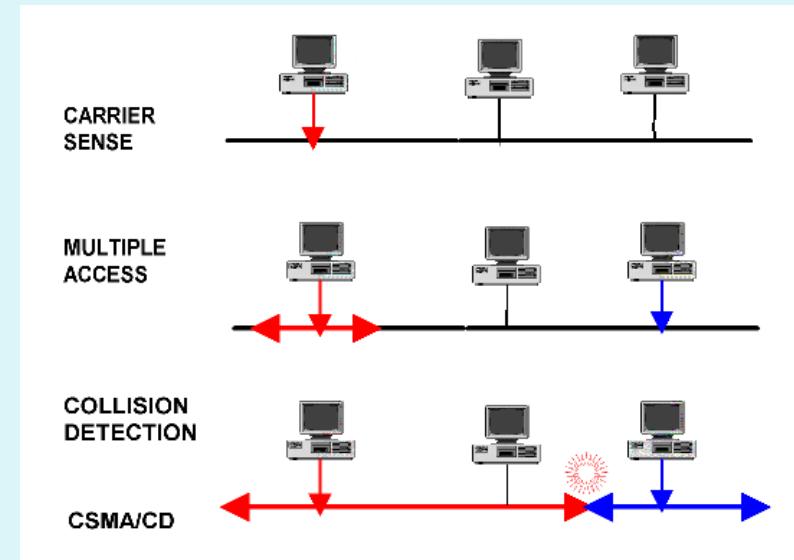


# Učinkovitost ALOHA in razsekana ALOHA

- učinkovitost: delež časa, v katerem je bilo oddajanje uspešno
- ALOHA:
  - nizka prepustnost (18%)
- razsekana ALOHA:
  - boljša prepustnost (37%)
- upoštevanje smernic za *idealni protokol*:
  1. če oddaja eno samo vozlišče, oddaja s hitrostjo R 
  2. če oddaja M vozlišč, oddajajo s povprečno hitrostjo  $R/M$  
  3. enostavnost? (potrebna sinhronizacija)

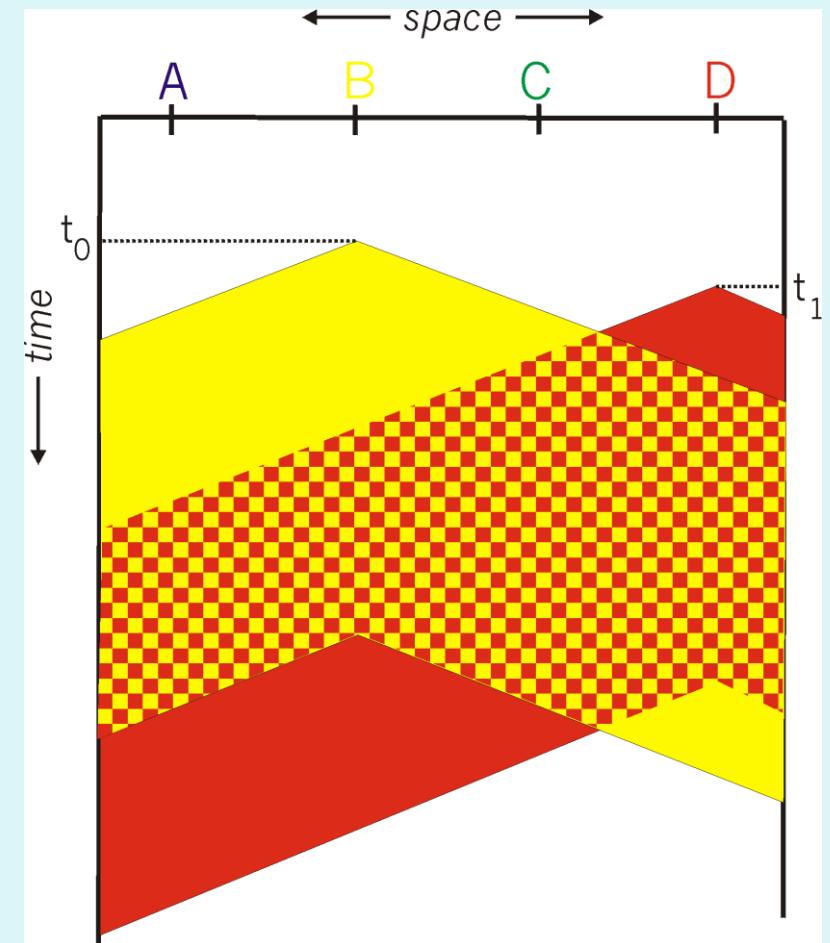
# CSMA

- **Carrier Sense Multiple Access**
  - vztrajni: če je kanal zaseden, posluša dokler se ne sprosti
  - nevztrajni: šele po času ponovno prisluhne
  - p-vztrajni: vztrajno posluša, ko se kanal sprosti, z verjetnostjo p odda paket
- princip:
  - poslušaj, ali že kdo govori, preden spregovoriš (zaznavaj druge pošiljatelje)



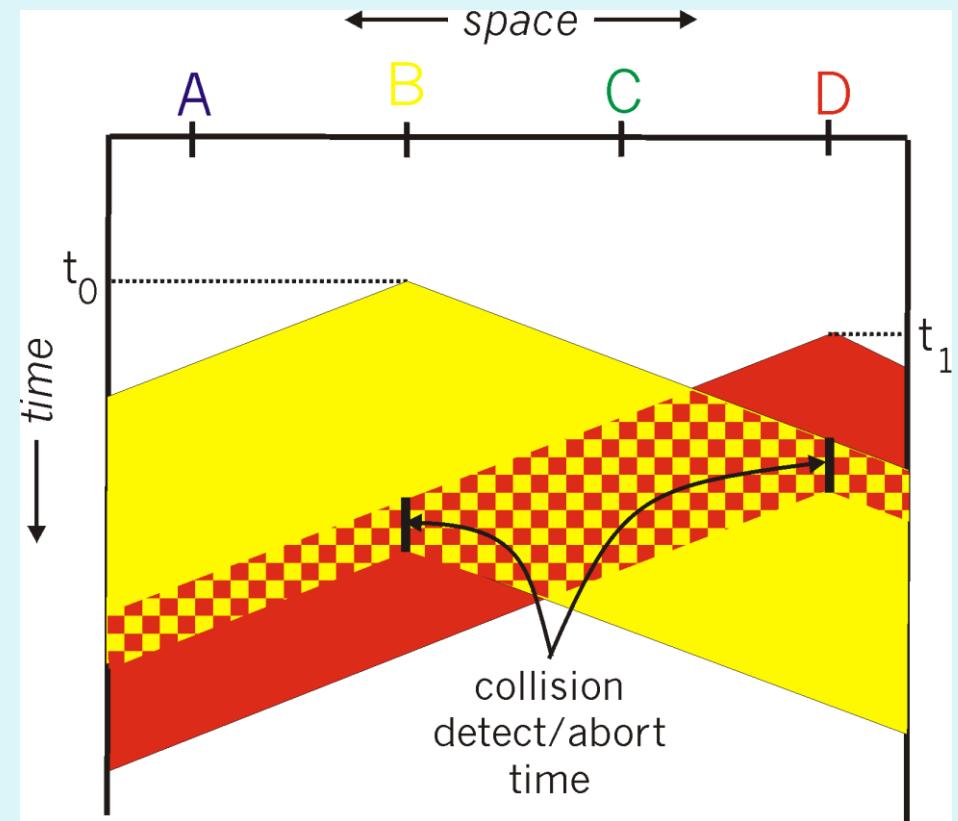
# CSMA

- zakaj je sploh potrebno zaznavanje kolizij, če velja princip poslušanja obstoječih govorcev?
  - odgovor: zaradi propagacijske zakasnitve signala skozi prostor medija
  - slika: CSMA



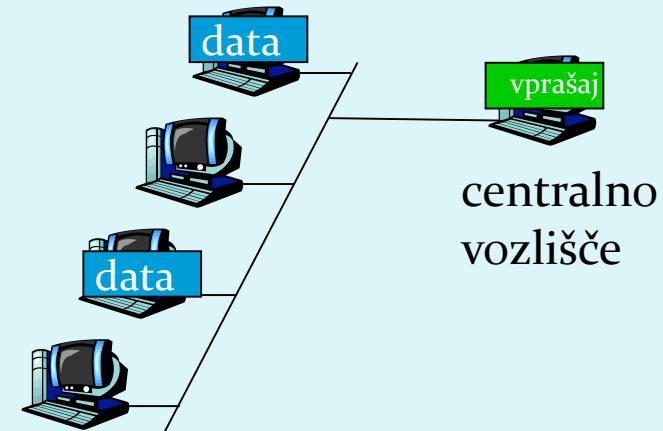
# CSMA/CD

- zaznavanje kolizij (CD) omogoči prekinitve komunikacije v primeru kolizije in s tem hitrejšo sprostitev kanala
- princip: če vskočiš komu v besedo, nehaj govoriti (prekini kolizijo)
- uporaba jam signala za obveščanje ostalih vozlišč, naj ne pošiljajo
- Primer: 802.3 Ethernet



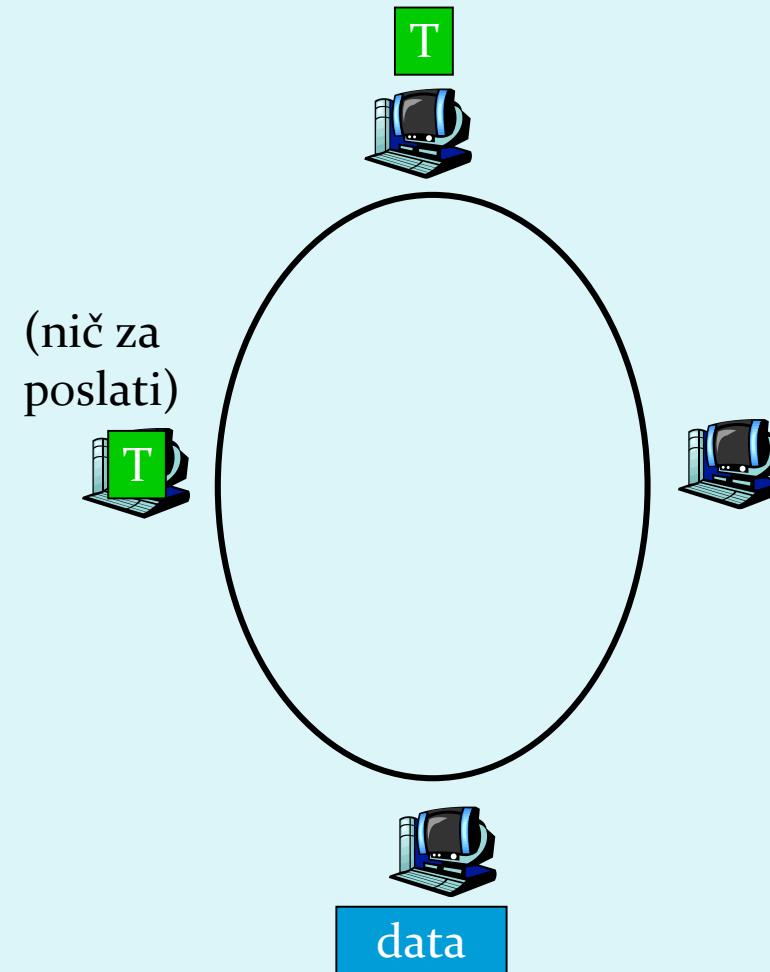
### 3. Protokoli za izmenični dostop

- Namesto faze boja za medij je faza rezervacije. Dva pristopa:
  1. rezervacija s centralnim vozliščem (polling)
    - zakasnitev
    - enotna točka odpovedi (centralno vozlišče)



# 3. Protokoli za izmenični dostop

2. rezervacija z žetonom, ki kroži (token)
  - zakasnitev
  - enotna točka odpovedi (žeton)



### 3. Protokoli za izmenični dostop

- primer: Vodilo in obroč z žetonom (FDDI, Token Ring)
- upoštevanje smernic za idealni protokol (najboljša kombinacija):
  1. če oddaja eno samo vozlišče, oddaja s hitrostjo R 
  2. če oddaja M vozlišč, oddajajo s povprečno hitrostjo  $R/M$  
  3. vendar: koordinacija žetonov doprinaša k zakasnitvi 

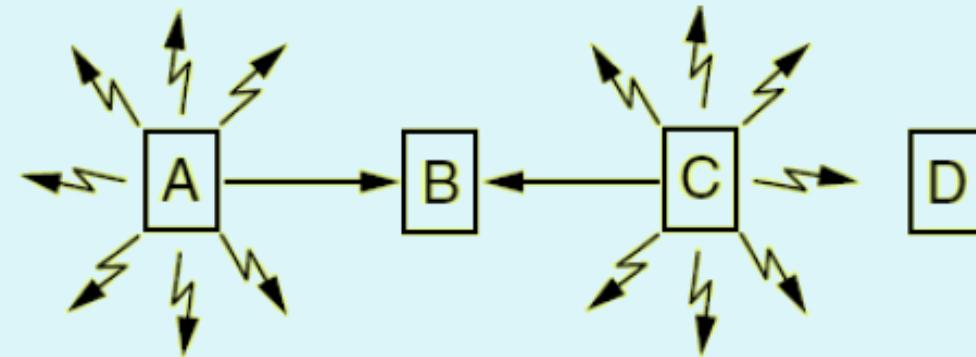
# Storitve povezavne plasti

- povezavna plast izvaja:
  1. **okvirjanje datagramov:** podatkom višje plasti se doda glava in določi struktura 
  2. **zaznavanje in odpravljanje napak:** z dodatnimi biti lahko zaznavamo, ali je prišlo do napake pri prenosu okvirja; v določenih primerih jo lahko odpravimo 
  3. **dostop do medija:** če je medij deljen, se uporablja MAC protokol (*media access control*) in ustrezeno naslavljjanje udeležencev 
  4. **zagotavljanje zanesljive dostave:** uporaba potrjevanja in ponovnega pošiljanja v primeru napake pri prenosu na povezavi
  4. **kontrola pretoka:** usklajevanje hitrosti pošiljanja glede na procesorske sposobnosti prejemnika



# Dostop do brezžičnega medija

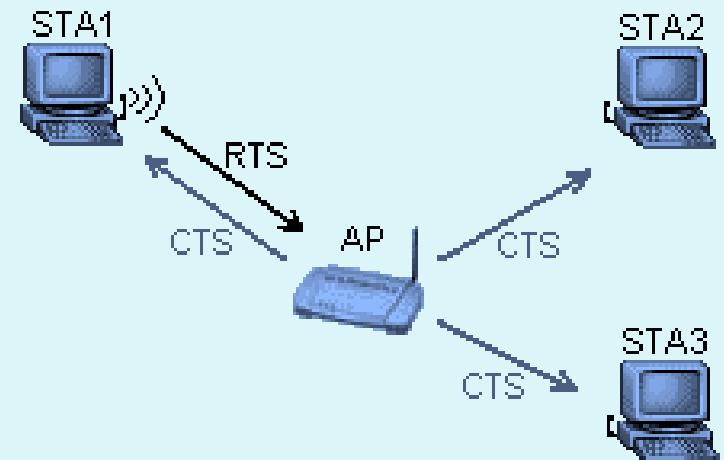
- CSMA/CA (Collision Avoidance): izogibanje trkom (kolizijam) z usklajevanjem časov pošiljanj
  - je tudi protokol za naključni dostop
  - CS ne deluje dobro, ker imajo dostopne točke različna področja pokritosti



- okvir se pošlje v celoti, kolizija se ne zaznava (CD), ker ni možno
- uporabljen v 802.11 (WiFi)

# Dostop do brezžičnega medija

- uporabljen v 802.11 (WiFi)
- uporaba signalov za rezervacijo medija
  - RTS (Request To Send): "želim pošiljati"
  - CTS (Clear To Send): "izvoli, pošiljaj"

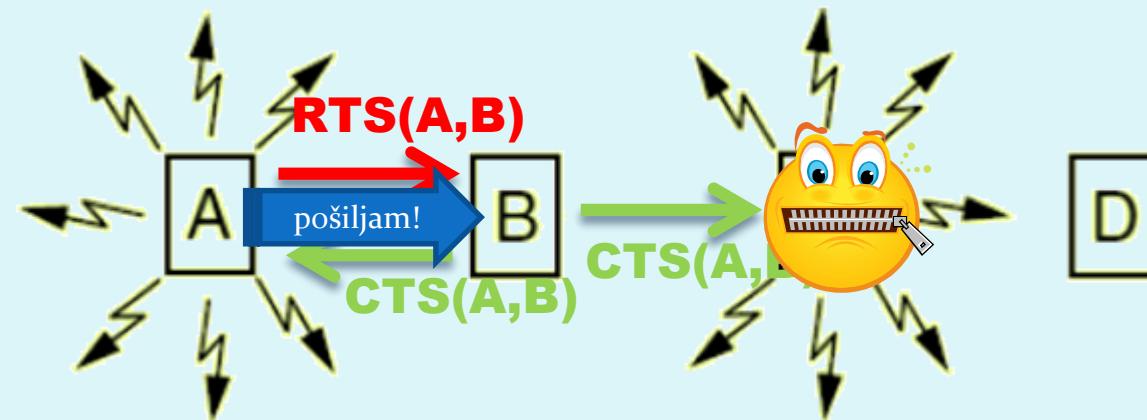
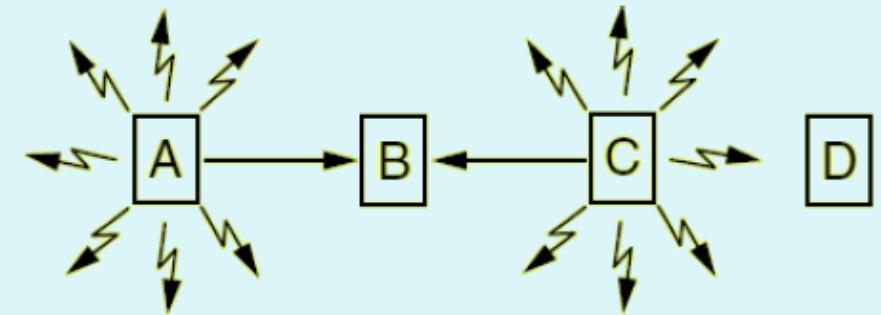


# Dostop do medija: 802.11 WiFi

Poznamo dve posebni situaciji

## 1. skriti terminali (A in C sta vzajemno skrita):

- terminala nista v dosegu (= sta skrita)
- terminala lahko vendarle ustvarita kolizijo v točki B
- kako pomagata RTS in CTS?

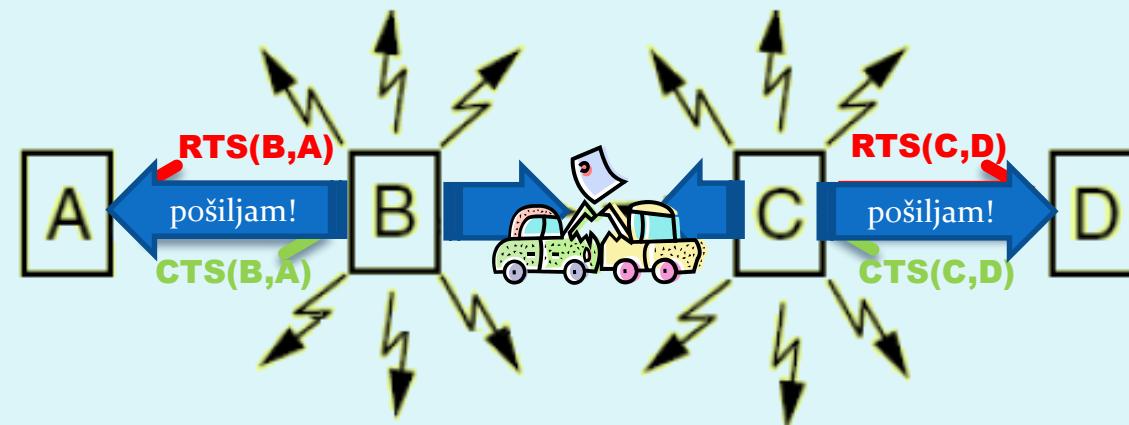
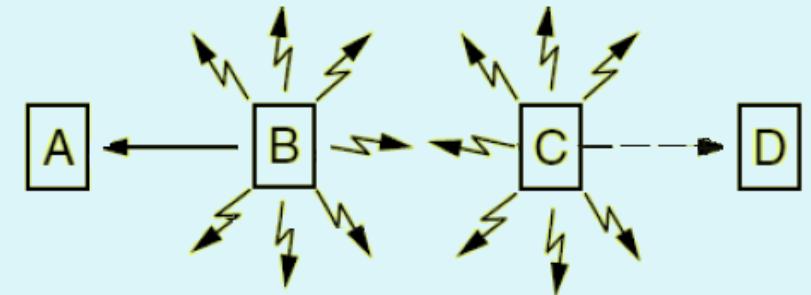


# Dostop do medija: 802.11 WiFi

Poznamo dve posebni situaciji

## 2. izpostavljeni terminali (B in C sta si izpostavljeni):

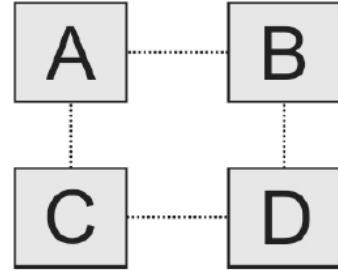
- terminala sta v dosegu (= sta izpostavljeni)
- terminala ustvarjata trke (kolizije) v vmesnem prostoru
- vendarle pa ne ustvarjata trkov pri pošiljanju B->A in C->D



# Vaja

- notacija:  
" $A \rightarrow B$ " ali "*A pošilja B*" ali "*A pošlje okvir B*"  
pomeni:  
***"Ali se lahko en okvir uspešno prenese od pošiljatelja A do prejemnika B"***  
(ne zahtevamo, da B nanj tudi odgovori ali da se vzpostavi daljša komunikacija)
- (pozor, **popravek** strani 53 v natisnjem učbeniku)

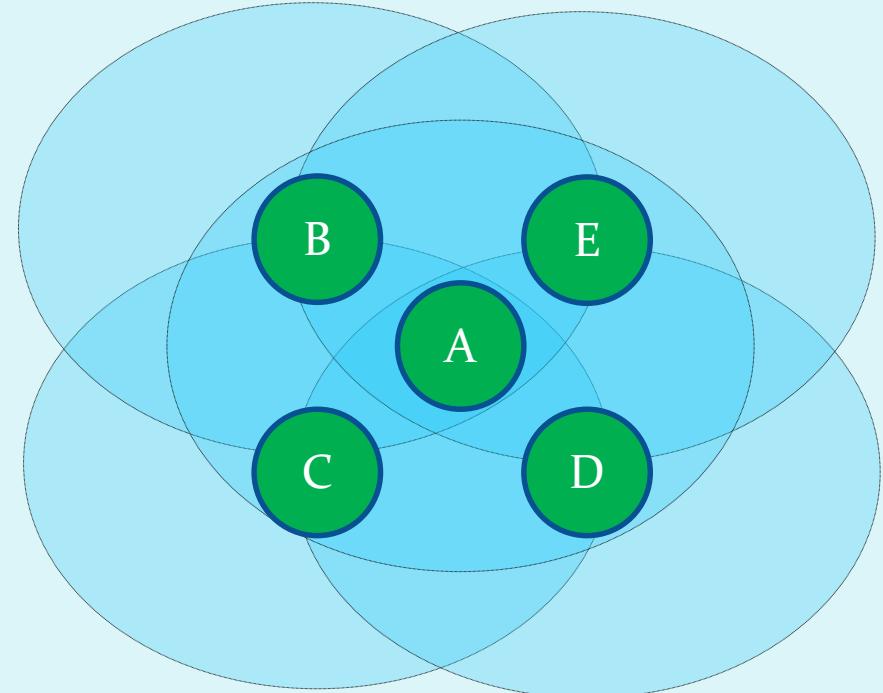
★ **Primer:** Za vajo analizirajmo sistem postavitve brezžičnih omrežnih vozlišč, ki je prikazan na spodnji sliki (doseg med vozlišči je označen s črtkanimi povezavami namesto z elipsami). Slike vidimo, da sta v dosegu vsakega izmed štirih vozlišč dve preostali vozlišči. Zastavimo si in odgovorimo na dve vprašanji:



- Ali lahko medtem, ko vozlišče A pošilja vozlišču B, tudi vozlišče D pošilja vozlišču C? *Odgovor je ne.* Namreč, če vozlišče A pošlje RTS vozlišču B, bosta CTS vozlišča B slišala vozlišči A in D. D zato v času, ko A pošilja, ne bo oddajal.
- Ali lahko medtem, ko vozlišče A pošilja vozlišču B, tudi vozlišče C pošilja vozlišču D? *Odgovor je da.* CTS, s katerim odgovori vozlišče B vozlišču A, slišita samo vozlišči A in D, ne pa tudi vozlišče C. Vozlišče C lahko zato vozlišču D okvir, s katerim ne bo zmotil B-jevega sprejema, kljub temu, da sta si vozlišči A in C izpostavljeni.

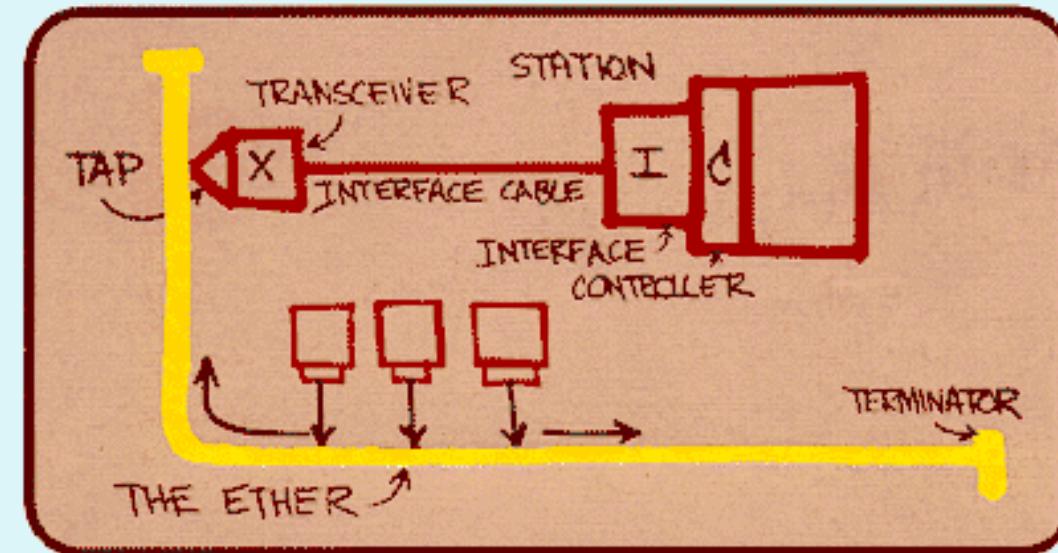
# Primer

- optimizirati želimo pošiljanje. Situacije:
  - A je v dosegu B, C, D in E
  - B je v dosegu z A, C in E
  - C je v dosegu z A, B in D
  - D je v dosegu z A, C in E
  - E je v dosegu z A, B in D
- vprašanja:
  - kadar pošilja B  $\rightarrow$  C, ali lahko pošilja A  $\rightarrow$  D?
  - kadar pošilja B  $\rightarrow$  C, ali lahko E  $\rightarrow$  D?
  - kadar pošilja B  $\rightarrow$  A, ali drži, da ne more pošiljati nihče drug?
  - kadar pošilja A  $\rightarrow$  B, ali lahko C  $\rightarrow$  A?
  - kadar pošilja A  $\rightarrow$  B, ali lahko D pošilja komurkoli? (\*\*\*)



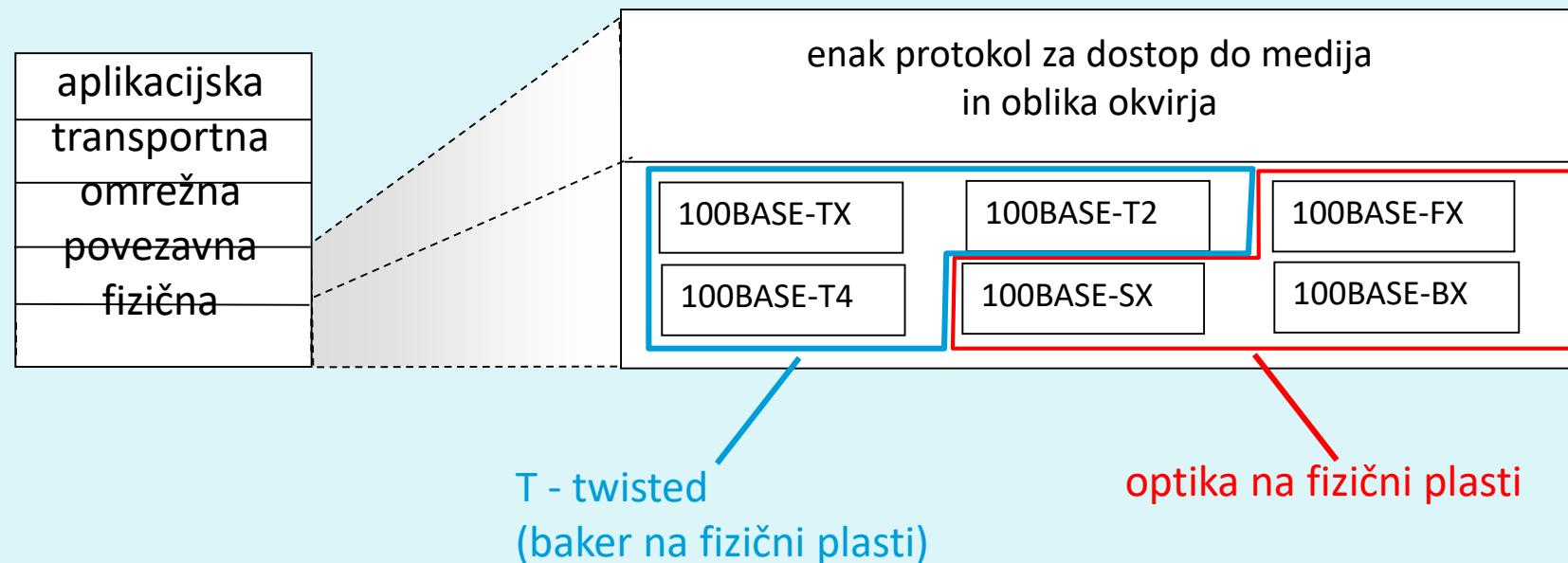
# Ethernet

- razvoj v 70. - 90. letih na podlagi token ring, FDDI, ATM
- razlogi za uspeh:
  - hiter (hitrost do 100 Gbps),
  - poceni,
  - enostavnejši v primerjavi z drugimi
- skica koncepta snovalca Etherjeta (Bob Metcalfe):



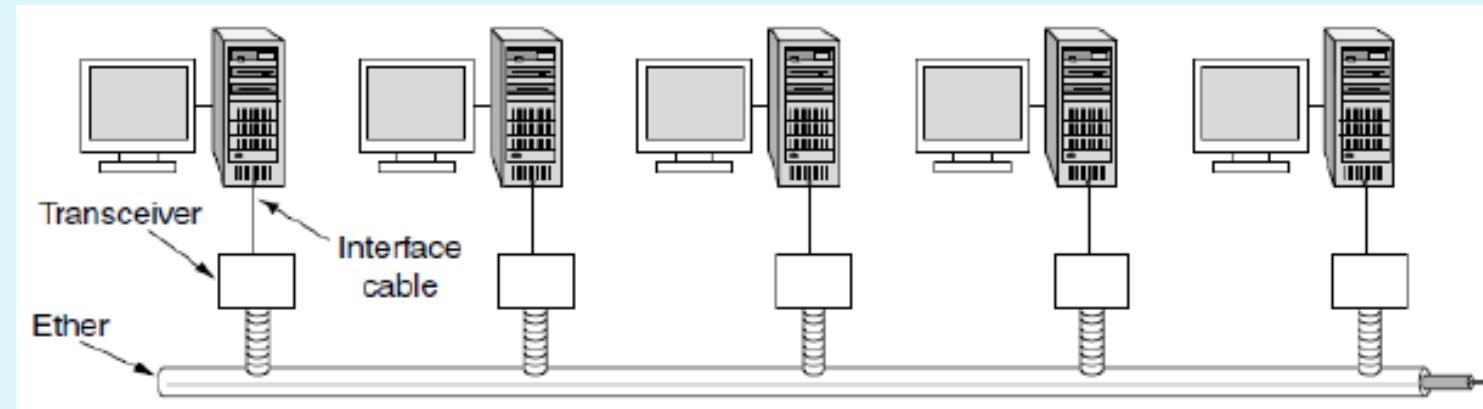
# Ethernet tehnologije

- različni **standardi** in **fizični mediji** (baker, optika), vsi pa imajo isto obliko okvirja in MAC protokol
- hitrosti: 2/10/100 Mbps, 1/10 Gbps
- oznaka: hitrost + BASE (osnovna frekvenca) + medij

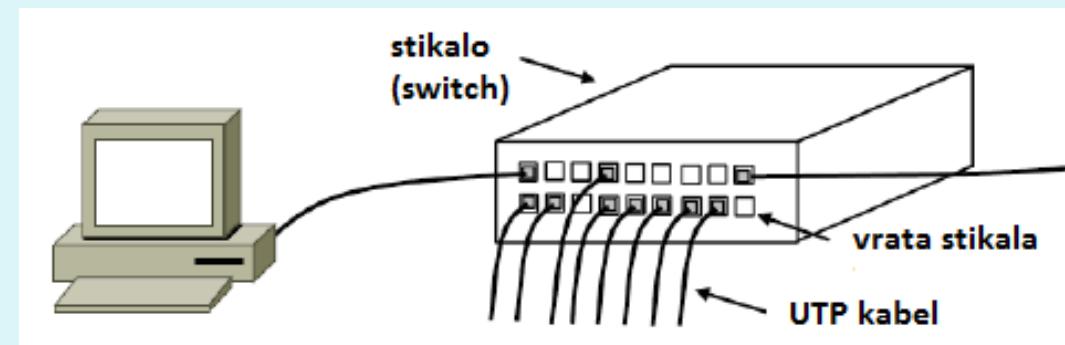


# Ethernet skozi čas

- zgodnji Ethernet

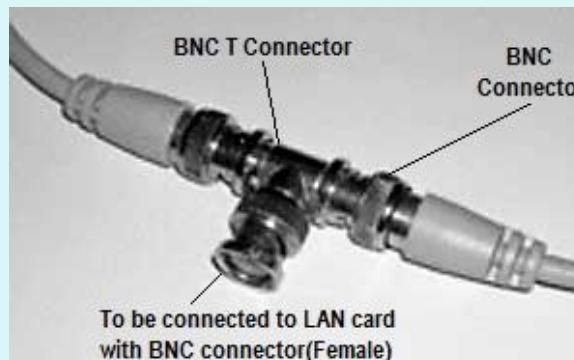
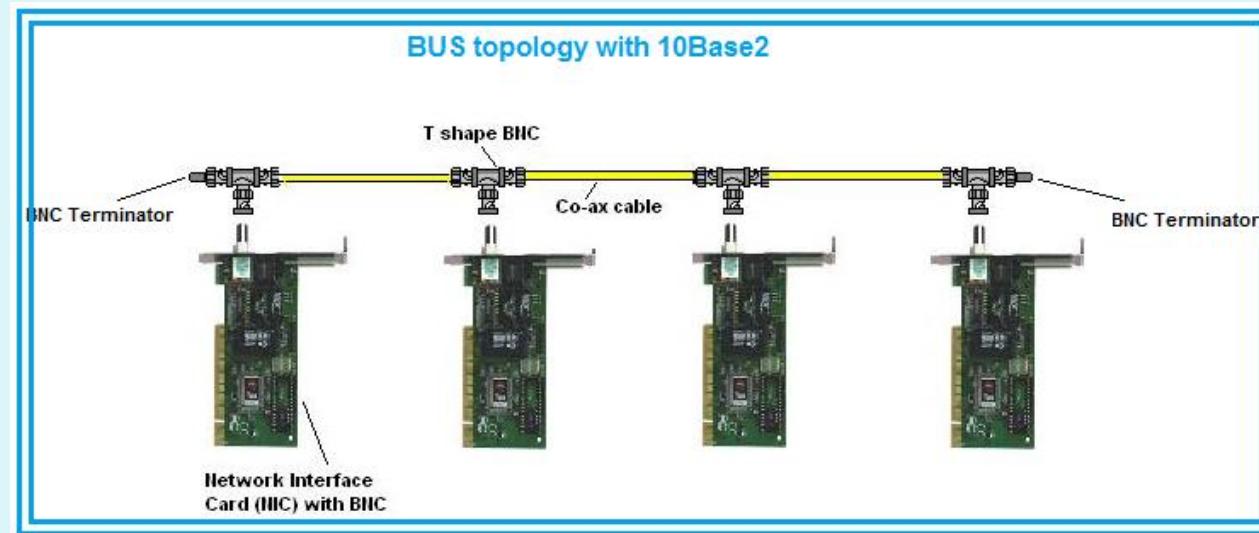


- sodobni Ethernet



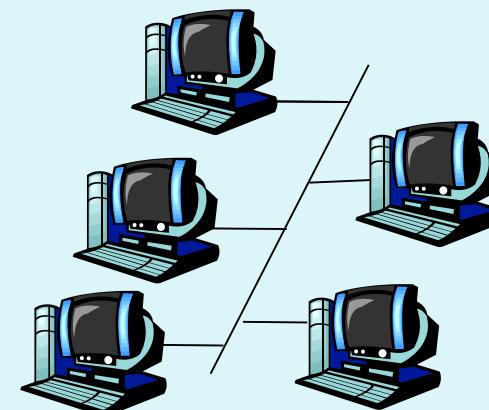
# Ethernet skozi čas

- vodilo

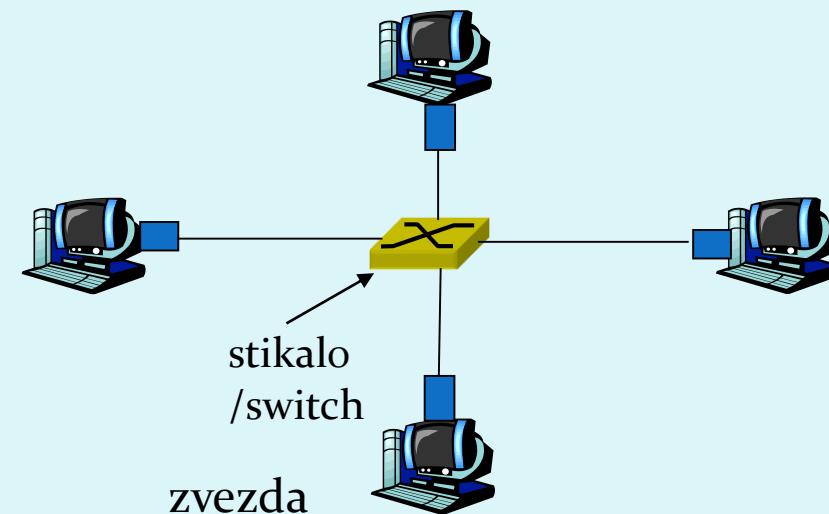


# Razvoj topologije Etherneta

- včasih: vodilo (bus)
  - vsi vmesniki v isti kolizijski domeni
  - najprej koaksialno vodilo (coax), nato topologija zvezda (naprava *hub* – razdelilec/zvezdišče)
- danes: zvezda (*stikalo* v centru) - okoli leta 2000
  - ločene kolizijske domene
  - možne različne različice Etherneta na vsakem kraku zvezde



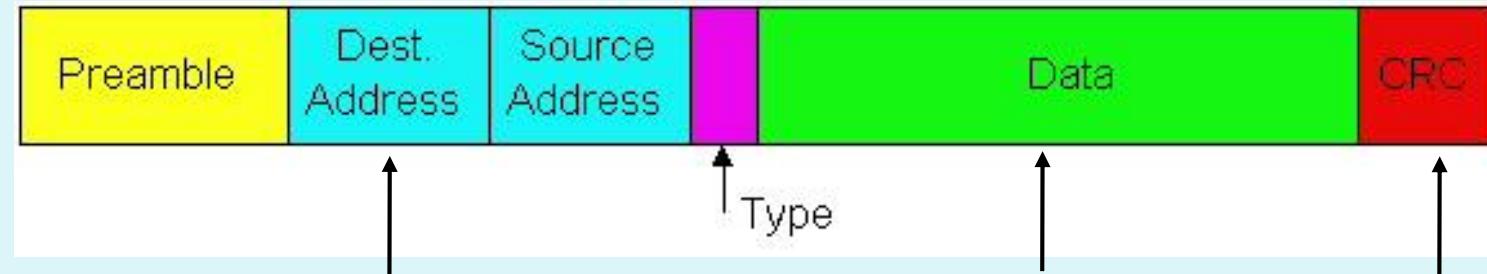
vodilo: koaksialni kabel



zvezda

# Okvir Ethernet

- Preamble:  $7 \times 10101010$  in  $1 \times 10101011$ 
  - namenjena sinhronizaciji ur oddajnika in prejemnika (in hitrosti)
  - znamenje: "pozor, sedaj prihajajo pomembni podatki"



6B: Če se ujema s  
prejemnikovim  
naslovom ali če je  
broadcast, ga  
posreduje omrežni  
plasti, sicer ga zavrže.

2B:  
Omrežni  
protokol  
(IP, ARP,  
Novell...)

Podatki – datagram  
dolžine  
46 – 1500B.  
MTU = 1500

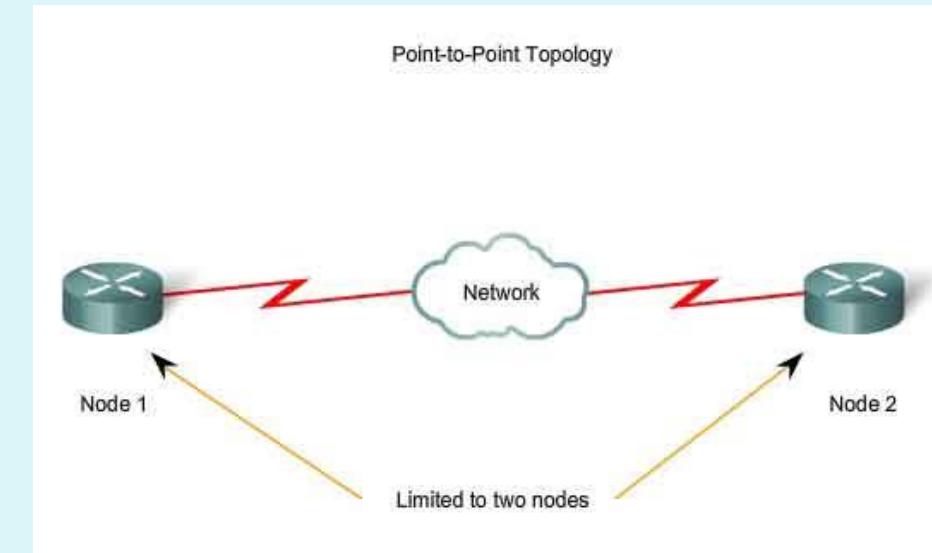
4B: Če ni  
ok,  
prejemnik  
okvir  
zavrže

# Kaj ponuja Ethernet?

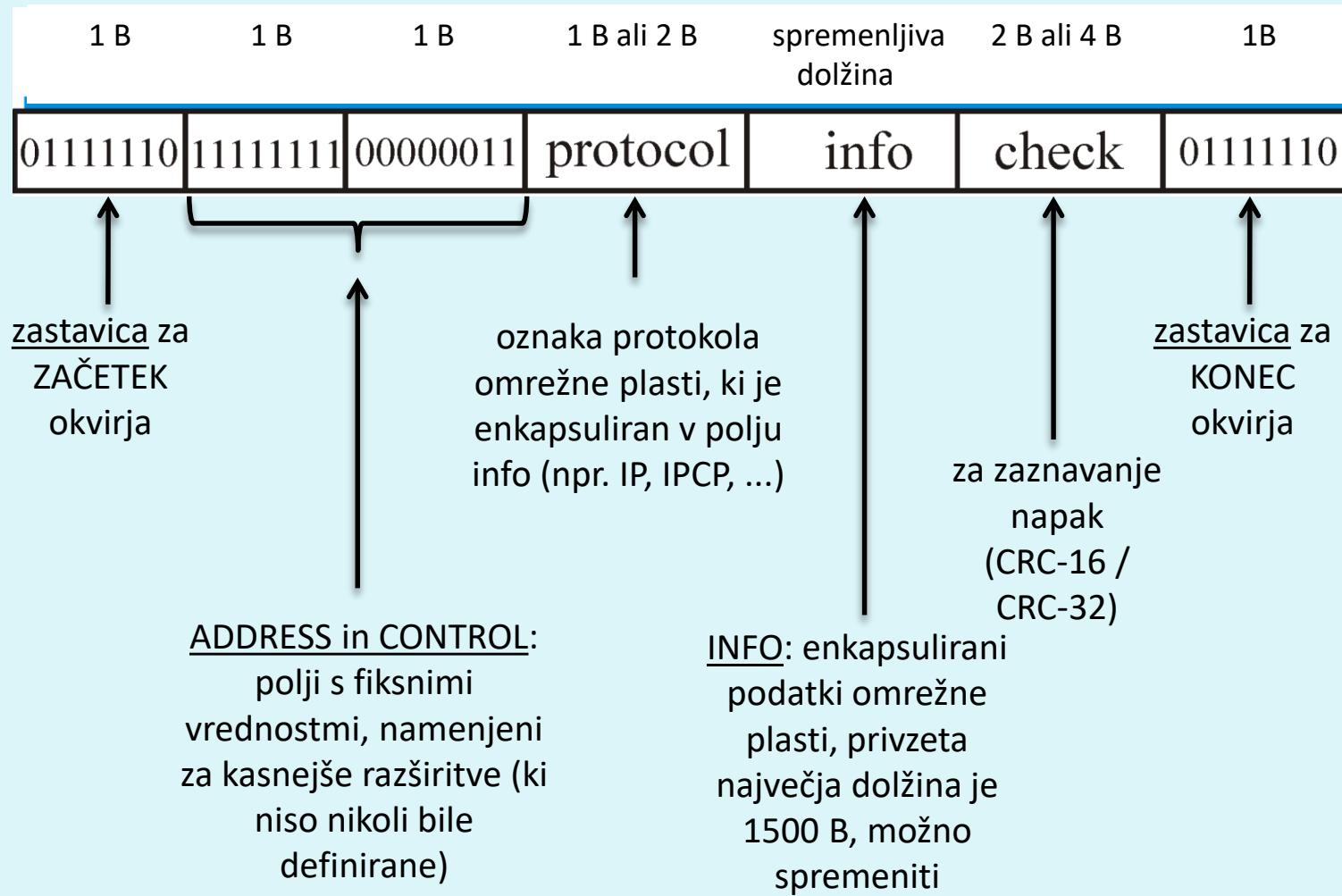
- **nepovezavna** storitev (ni rokovanja, vmesnik pošlje brez faze vzpostavljanja)
- **nezanesljiva** storitev
  - kontrola pravilnosti se izvaja s CRC
  - potrjevanje in ponovno pošiljanje se ne uporablja!
  - omrežna in transportna plast skrbita za vrstni red in prenos vseh podatkov
- **Ethernet uporablja CSMA/CD:** posluša pred oddajo, v primeru trka preneha
  - uporablja se *eksponentno povečevanje čakanja na naslednjo oddajo* (angl. *exponential backoff*):
    1. ob prvi koliziji pošlje po čakanju 0-1 okvirja
    2. ob drugi koliziji pošlje po čakanju 0-3 okvirje
    3. ob tretji koliziji pošlje po čakanju 0-7 okvirjev ...
  - pri koliziji uporablja t. i. **jam signal (motilni signal)** za oznanitev kolizije drugim (okvarimo CRC v drugih okvirjih, da dosežemo, da ostala vozlišča okvirje zavričejo)
  - **učinkovitost rabe medija 85 – 100 %!**

# Protokol PPP

- drugi povezavni protokol, poglejmo si alternativo
- PPP: Point-to-Point Protocol
- se ne uporablja na oddajnih (*broadcast*) povezavah, temveč na povezavah med dvema točkama (point-to-point)
- imamo enega pošiljatelja in enega prejemnika
- ni potrebe po protokolu za dostop do medija in MAC naslovih
- primer uporabe za: klicni dostop, ISDN, SONET/SDH, X.25

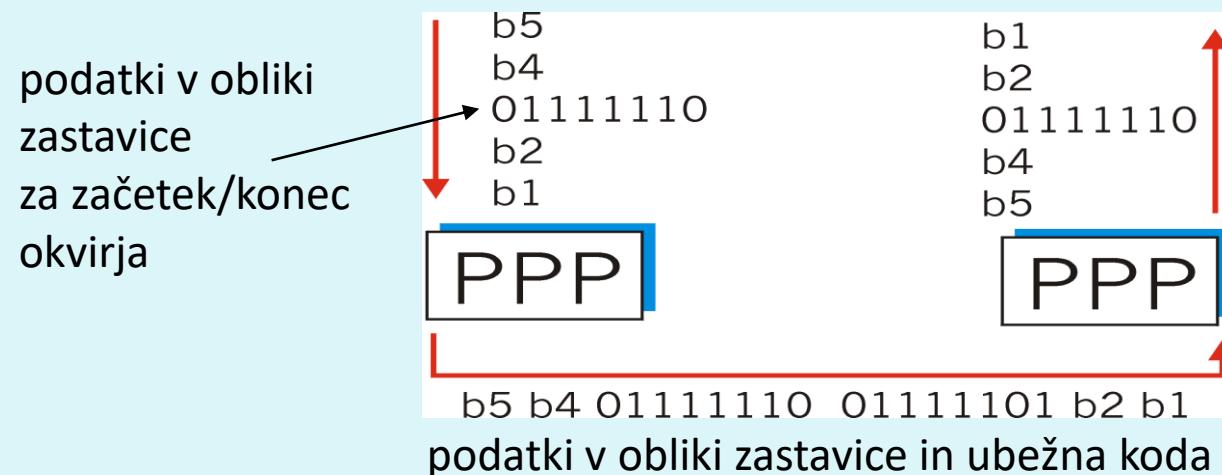


# Okvir PPP



# PPP: transparentnost podatkov

- želimo, da so podatki v okvirju PPP lahko poljubne oblike
  - kaj pa, če podatki vsebujejo niz 01111110, ki pomeni začetek/konec okvirja???
  - rešitev: uporabimo vrivanje (angl. *stuffing*)
  - postopek: pošiljatelj pred 01111110 vrine ubežno kodo (angl. escape sequence) 01111101, ki opozori prejemnika, da zaporedje, ki sledi, ni zastavica za začetek/konec okvirja; prejemnik ubežno kodo odstrani pri sprejemu



# Naslavljanje naprav na povezavni plasti

- Naprave imajo naslove, ki so sestavljeni iz **48 bitov = 6 B** (fizični naslov ali MAC<sup>[Media Access Control]</sup> naslov):
  - naslov zapišemo z 12 HEX znaki, npr.: 5C-66-AB-90-75-B1
  - možnih  $2^{48}$  naslosov, prva polovica je ID proizvajalca, druga polovica ID adapterja (podjetje zakupuje pakete po  $2^{24}$  naslosov)
  - MAC naslov je zapečen v adapterju (omrežni kartici) in je unikaten!



```
C:\WINDOWS\System32\cmd.exe
Windows IP Configuration

Host Name . . . . . : pb
Primary Dns Suffix . . . . . : helpdesk2.umd.edu
Node Type . . . . . : Unknown
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : helpdesk2.umd.edu
                                         umd.edu

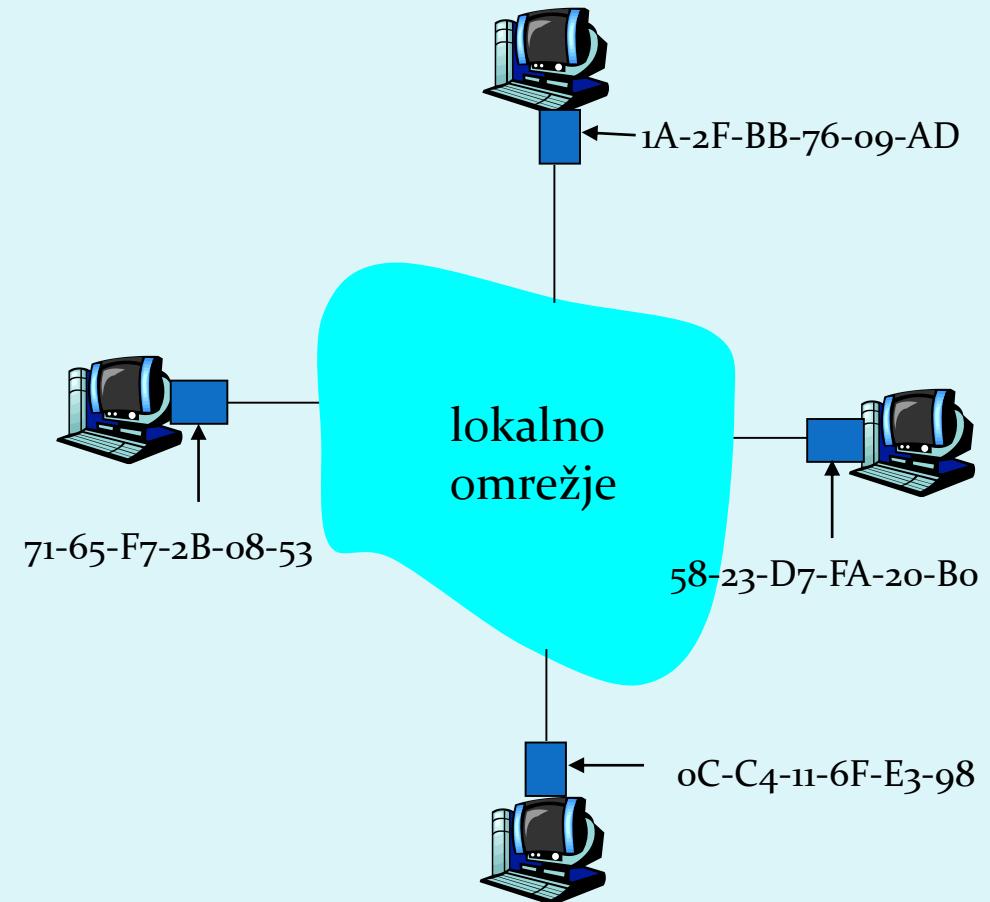
Ethernet adapter Local Area Connection:

  Connection-specific DNS Suffix . . . . . :
  Description . . . . . : 3Com 3C920 Integrated Fast Ethernet
  Controller <3C905C-TX Compatible> . . . . . :
  Physical Address . . . . . : 00-B0-D0-86-BB-F7
  Dhcp Enabled. . . . . : No
  IP Address . . . . . : 129.2.5.98
  Subnet Mask . . . . . : 255.255.254.0
  Default Gateway . . . . . : 129.2.4.1
  DNS Servers . . . . . : 128.8.74.2
                                         128.8.76.2

C:\Documents and Settings\Paul>
```

# Uporaba naslosov MAC

- naprave morajo razpoznati, ali je okvir na mediju namenjen njim
- namen naslova: naprave opazujejo okvirje na mediju in sprejmejo le tiste, ki so naslovljene na njih
- vsaka naprava ima unikaten naslov
- naslov FF-FF-FF-FF-FF-FF je poseben, pomeni "broadcast" (prejemniki so vse naprave)

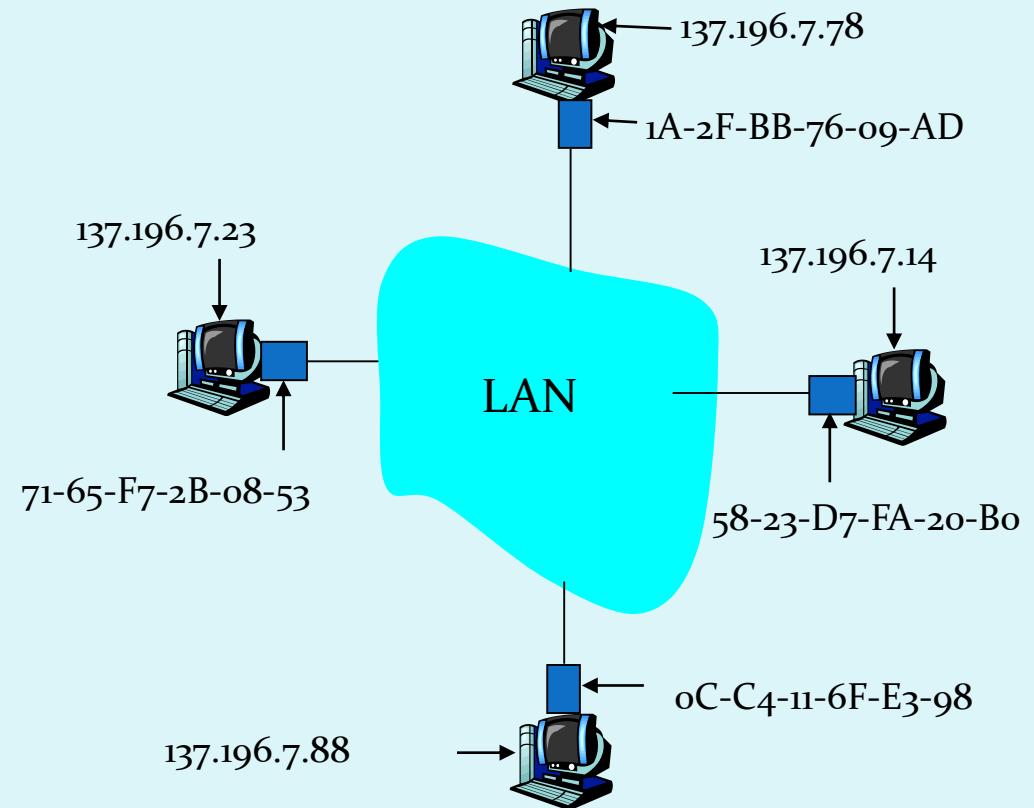


# Vendar pa ...

- (tega uradno še ne vemo...) naslavljanje računalnikov v Internetu poteka z uporabo **IP naslovov** (in ne MAC!), ki so na *omrežni plasti*:
  - **MAC naslovi** so **fizični** in **stalni** za napravo (podobno kot je EMŠO stalen),
  - **IP naslovi** so **logični** in **zamenljivi** oziroma odvisni od lokacije priklopa (podobno kot je poštni naslov spremenljiv)
- **primer:** naprava A želi poslati podatke napravi B. Za pošiljanje jo naslovi z naslovom IP naprave B in pred datagram povezavni plasti, ki je zadolžena, da jo dostavi. Povezavna plast mora napravo B nasloviti z naslovom MAC. Kako priti do njega?
  - Potrebujemo ARP: *Address Resolution Protocol!*

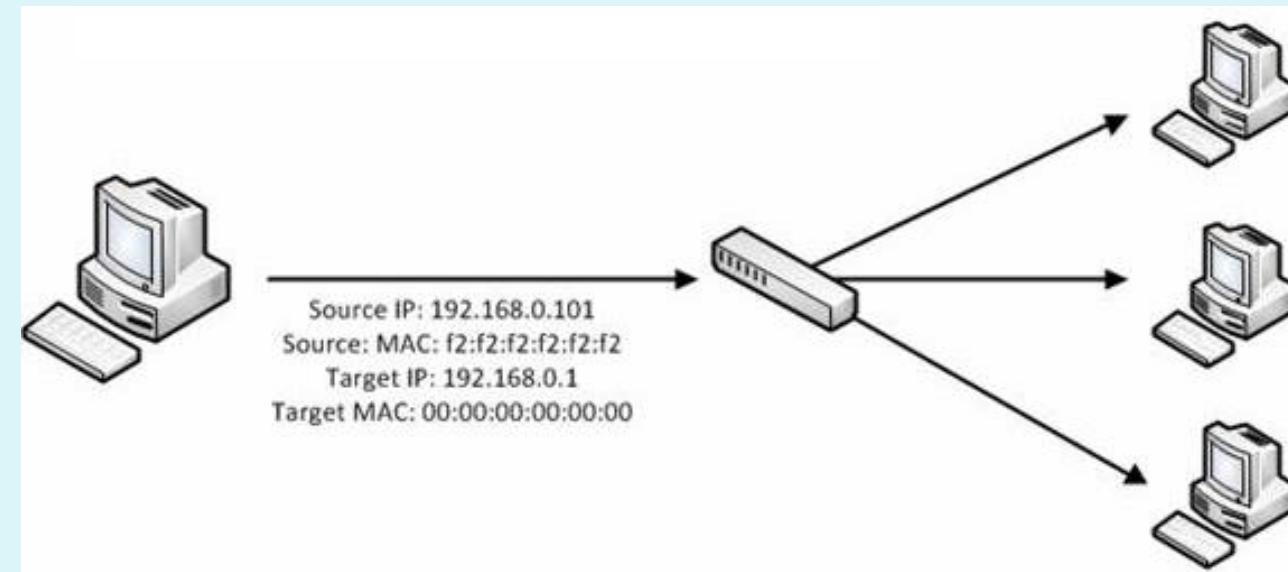
# ARP: Address Resolution Protocol

- vsako vozlišče ima tabelo ARP (preslikovalna tabela naslovov IP v naslove MAC ), ki vsebuje 3 podatke:  
<naslov IP | naslov MAC | TTL (20 min)>
- ARP deluje samo na lokalnem podomrežju in ne v celiem internetu
- Kako se tabela ARP napolni?
  - uporaba paketov ARP query



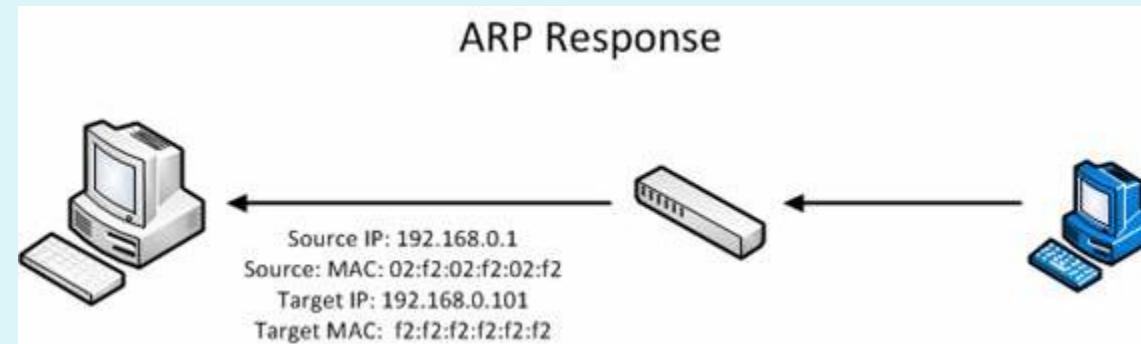
# ARP: nastanek tabele ARP

- A želi poslati datagram vozlišču B, od katerega naslova MAC še nima v tabeli ARP. A naredi naslednje:
  1. A pošlje paket za **ARP poizvedbo** (ARP request/query), ki ga naslovi na vse vmesnike (broadcast, FF-FF-FF-FF-FF-FF) in ki vsebuje naslov IP vozlišča B  
**"Kakšen je MAC naslov za napravo z naslovom IPxxx?"**
  2. ARP poizvedbo prejmejo vsa vozlišča, priključena na medij  
*(... nadaljevanje...)*



# ARP: nastanek tabele ARP

3. iz naslova IP v poizvedbi vmesnik B zazna, da paket sprašuje po njem
4. B odgovori vmesniku A z **odgovorom ARP** (*ARP response*), v katerem pošlje svoj naslov MAC (pri tem naslovi samo vozlišče A).
5. A shrani novi podatek v svojo tabelo



- tabela ARP nastaja torej po principu plug&play
- Analogija: Profesor zavpije v predavalnici:  
"Kakšen je EMŠO študenta, ki sedi na tretjem sedežu v drugi vrsti?"



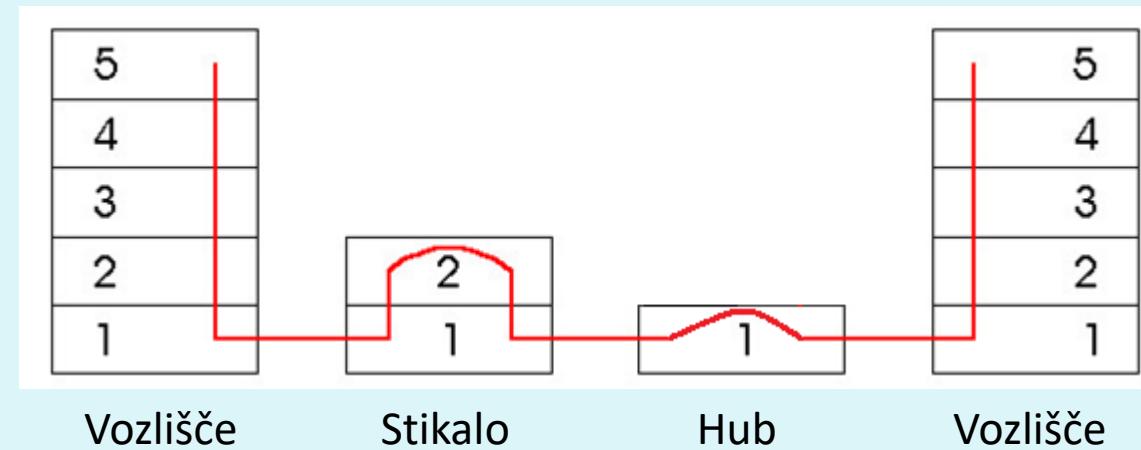
# Aktivna oprema (naprave)

- za povezovanje omrežne opreme se uporabljojo:
  - **repeater**: ponavljalec (ojačevalec) signala na fizični plasti
  - **hub**: razdelilec/zvezdišče, ponavlja signal na vseh ostalih vratih (ista hitrost), ne shranjuje okvirjev, ista kolizijska domena
  - **switch**: omrežno stikalo, *preklaplja* med priključenimi segmenti, shranjuje okvirje in aktivno ukrepa na podlagi vsebine
    - transparentni uporabiku, plug-and-play
    - opravlja **posredovanje, poplavljjanje in filtriranje**



# Primerjava

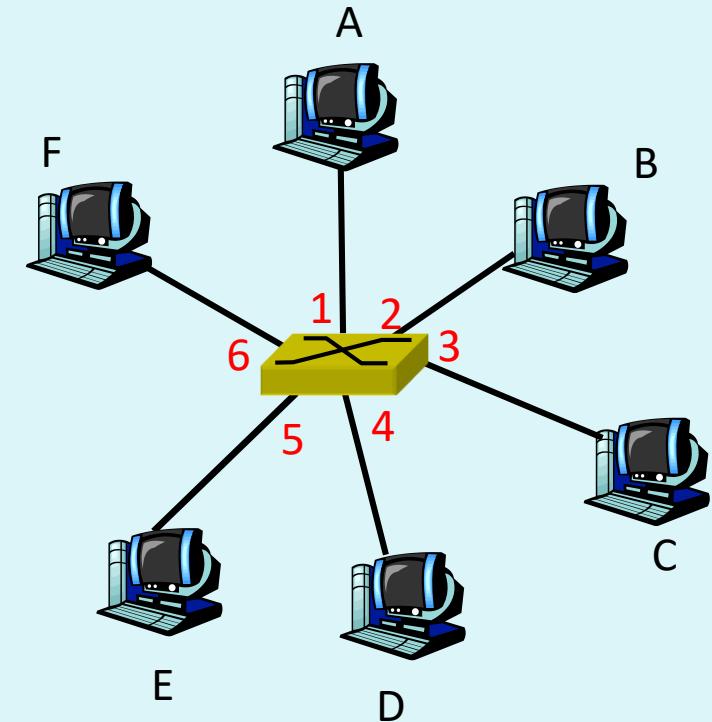
	Hub	Stikalo
Izolacija prometa	Ne	Da
Potrebna konfiguracija?	Ne	Ne



# Stikalo

- dopušča več **hkratnih** povezav (npr. A-F in B-C) in **dvosmerno** (full duplex) povezavo z vozlišči
- stikalo uporablja stikalno tabelo (**switch table**, **CAM table**, **FIB**), da se odloči, na katera vrata poslati okvir:
  - < naslov MAC | vrata do vmesnika | čas >

naslov MAC	vrata	TTL



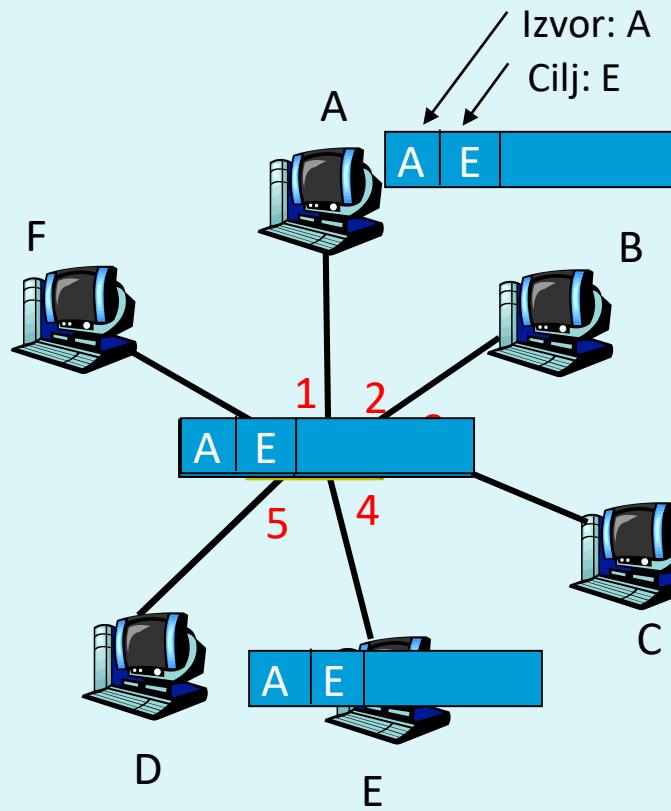
stikalo s 6 vmesniki

# Uporaba stikalne tabele

- stikalo se po priklopu uči, kje je dosegljiv kateri vmesnik, in samo vnaša zapise v stikalno tabelo
- kadar stikalo sprejme okvir, si za nekaj časa zapomni **lokacijo pošiljatelja** okvirja (=učenje)
- različne akcije pri sprejemu okvirja:
  - **POPLAVLJANJE na VSA vrata** (flooding, če ne vemo, kje je prejemnik)
  - **POSREDOVANJE na IZBRANA vrata** (če vemo, kje je prejemnik)
  - **FILTRIRANJE** (okvir je namenjen istim vratom, zavrzemo ga)

# Prvi sprejem okvirja

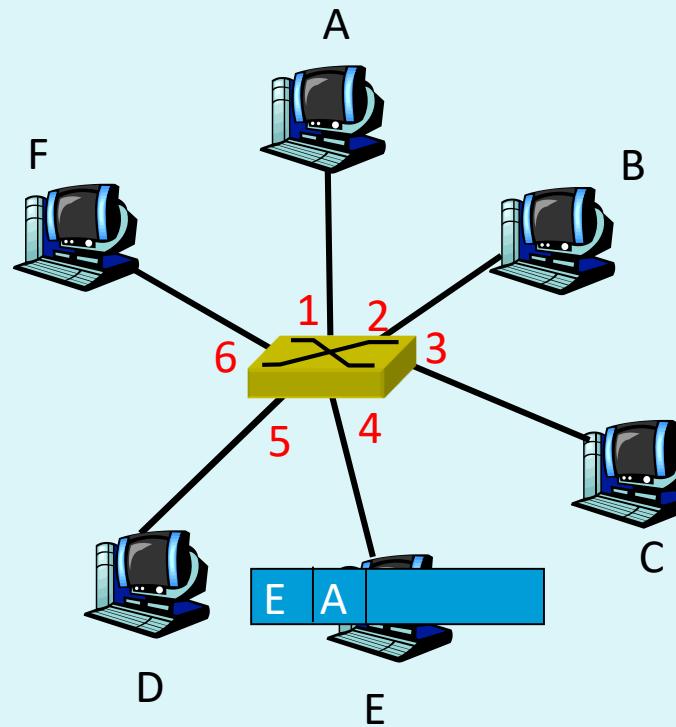
- stikalo **shrani lokacijo** pošiljatelja A
- ker stikalo ne ve, kje je cilj, **poplavi** (flood) okvir na vsa vrata



naslov MAC	vrata	TTL
A	1	60

# Pošiljanje znanemu prejemniku

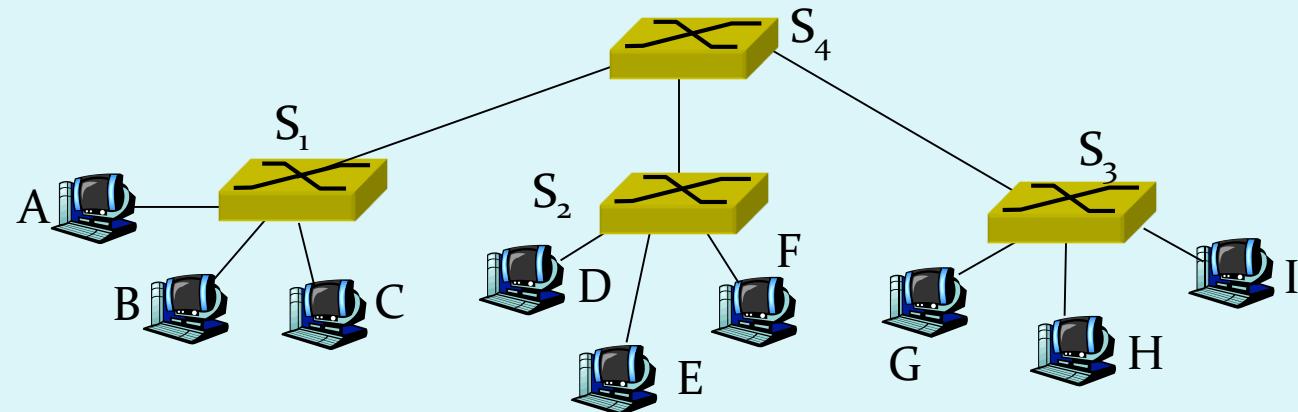
- stikalo prejme okvir za **znanega prejemnika (A)**
- ker stikalo ve, kje je cilj, **posreduje** okvir na prava vrata (1)



naslov MAC	vrata	TTL
A	1	60
E	4	60

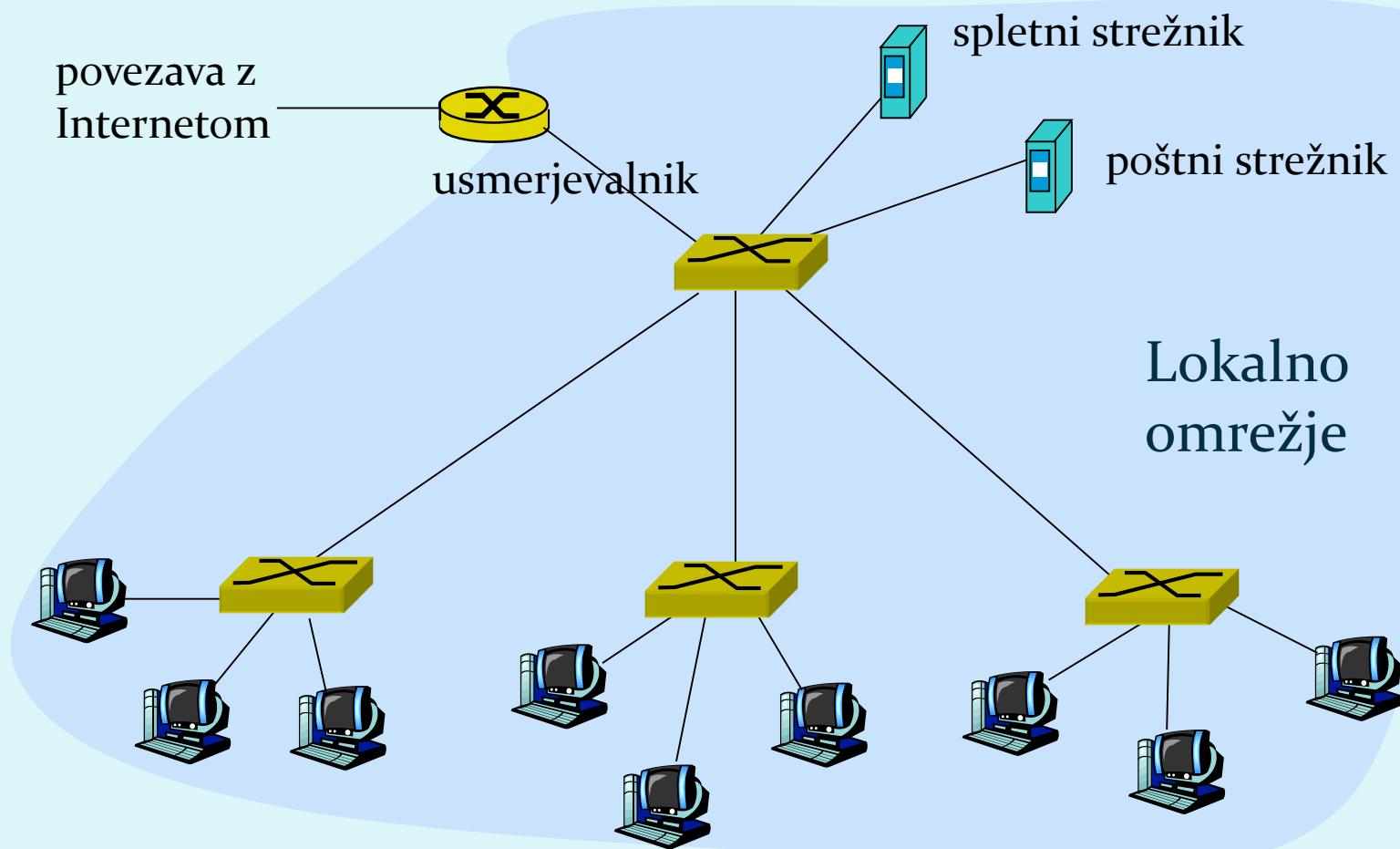
# Povezovanje stikal

- stikala lahko medsebojno povežemo
- pravilnega posredovanja se naučimo z učenjem stikalnih tabel



- **FILTRIRANJE:** stikalo  $S_4$  bo zavrglo okvir, ki ga npr. A pošlje C, ker je na istih vratih (če ima stikalo  $S_4$  v stikalni tabeli zapis za C)

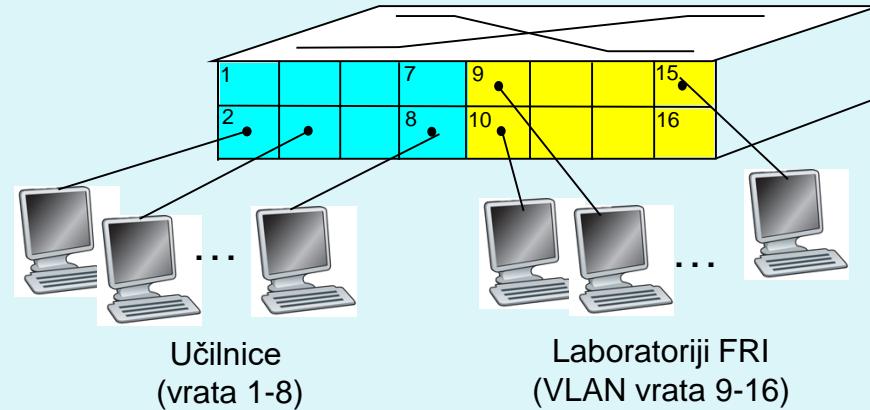
# Primer uporabe stikal



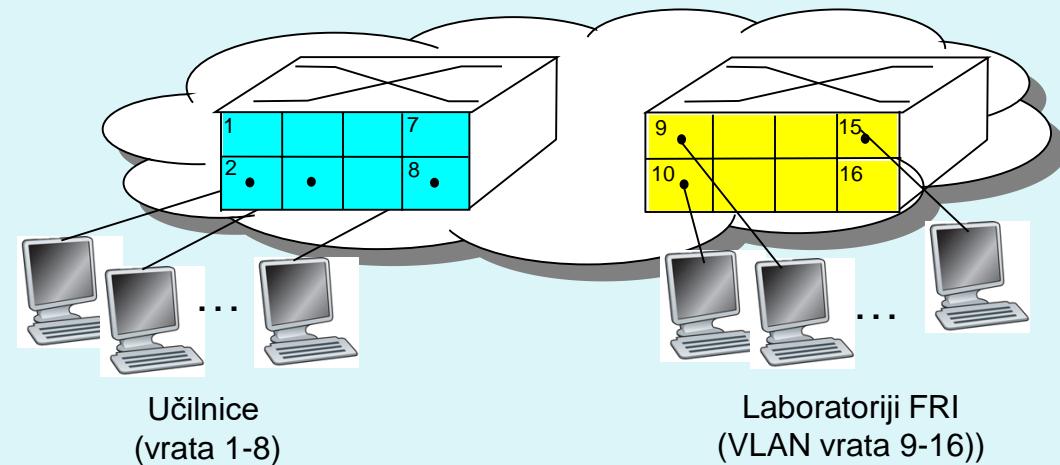
# Virtualna lokalna omrežja (VLAN)

- Uporaba stikal znotraj podjetja ima naslednje slabosti:
  - uporaba enega samega stikala kliče k pomanjkanju izolacije prometa (zmanjšanje broadcast prometa lahko izboljša performanse, varnostni/zasebnostni razlogi)
  - uporaba več stikal je cenovno draga rešitev
  - premik uporabnika na drugo lokacijo zahteva fizično vzpostavitev povezave
- Rešitev: uporaba virtualnih lokalnih omrežij (Virtual Local Area Network). Stikalo, ki podpira VLANe **omogoča uporabo različnih navideznih lokalnih omrežij na isti fizični omrežni infrastrukturi** (= delitev omrežja na več navideznih podomrežij)

# Virtualna lokalna omrežja (VLAN)



zgornje deluje enako kot več posameznih stikal



# Naslednje poglavje?

- omrežna plast

