

# Računalniške komunikacije

## 2019/20

povezavna plast  
naslavljanje, Ethernet,  
PPP, ARP, stikala

# Pridobljeno znanje s prejšnjih predavanj

- naloge/**storitve povezavne plasti**: okvirjanje, zaznavanje/odpravljanje napak, dostop do medija, zanesljiva dostava, kontrola pretoka
- **okvirjanje**
  - okvir
  - enkapsulacija/dekapsulacija po plasteh komunikacijskega modela
- **zaznavanje in odpravljanje napak**
  - razlogi za nastanek napak
  - odpravljanje z dodatnimi biti (EDC)
  - pariteta (1D/2D, soda/liha), Hammingova koda, CRC
- **dostop do medija s protokoli MAC**
  - vrste skupinskih medijev
  - **trk** (kolizija)
  - kriteriji za vrednotenje protokolov MAC (izkoristek, pravičnost, decentraliziranost, enostavnost)
  - družine protokolov:
    - **delitev kanala**: TDMA, FDMA
    - **naključni dostop**: ALOHA, razsekana ALOHA, CSMA, CSMA/CD
    - **izmenični dostop**: centralno vozlišče, krog z žetonom

# Storitve povezavne plasti

Povezavna plast izvaja:

1. **okvirjanje datagramov:** začetek/konec okvirja, glava/rep
2. **zaznavanje in odpravljanje napak:** z dodatnimi biti (EDC)
3. **dostop do medija:** protokol (MAC) in naslavljanje udeležencev
4. **zagotavljanje zanesljive dostave:** potrjevanje in ponovno pošiljanje
5. **kontrola pretoka:** izogibanje prekoračitvi medpomnilnika

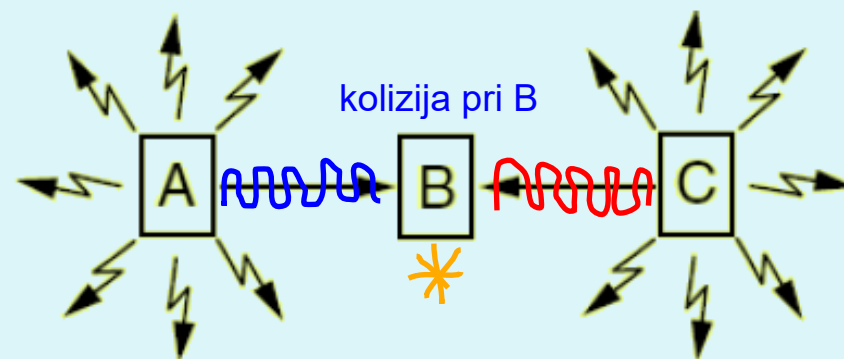
protokoli, ki jih bomo obravnavali, tega ne delajo (razen TCP na transportni plasti)

# 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

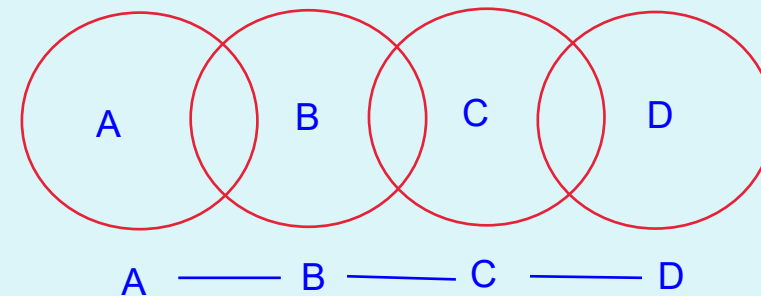
A ima v dosegu samo B,  
B ima v dosegu A in C,  
C ima v dosegu samo B ...



Primer:

- A pošilja B
- C želi uporabiti CS
- C posluša, če kdo govori, ampak A ni v njegovem dosegu, zato ga ne sliši
- C začne pošiljati B -> kolizija

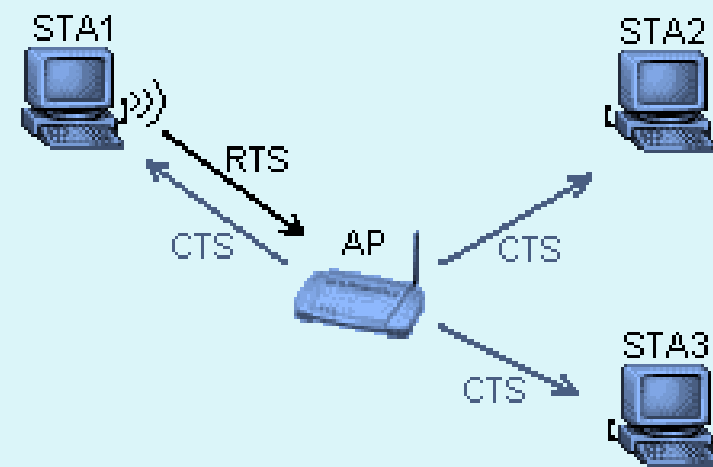
različne sheme (kako ponazorimo omrežje):



- 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 (to so navadni okvirji)
  - RTS (Request To Send): "želim pošiljati"
  - CTS (Clear To Send): "izvoli, pošiljaj"



# Dostop do medija: 802.11 WiFi

## Poznamo dve posebni situaciji

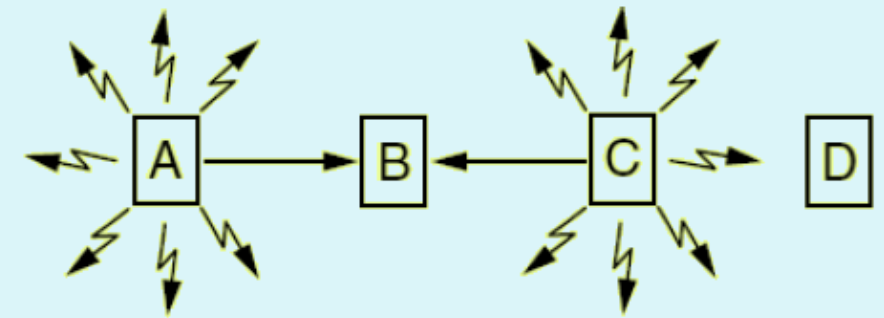
(hidden terminals)

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

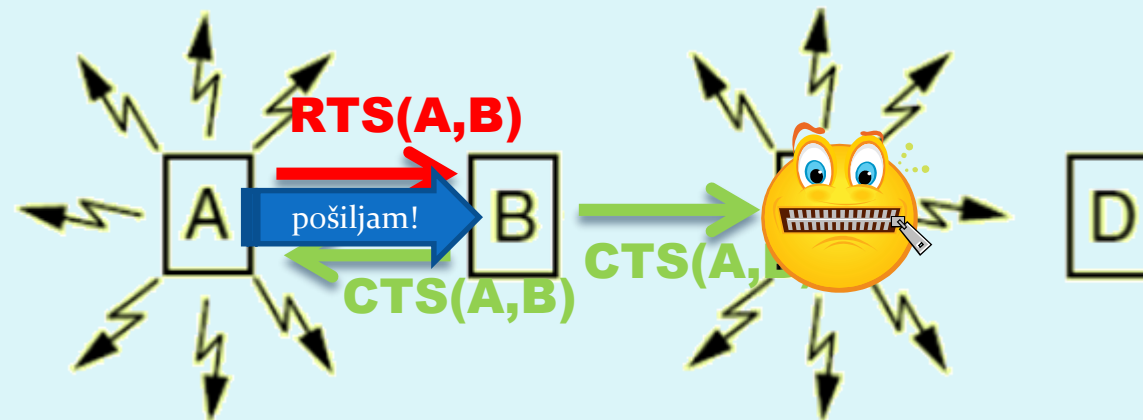
- terminala nista v dosegu (= sta skrita)
- terminala lahko vendarle ustvarita kolizijo v točki B
- kako pomagata RTS in CTS?

(gledamo zaporedja treh postaj)

A vidi B, B vidi A in C, C vidi B in D, D vidi C  
→ A ne vidi C itd.



signal se širi po prostoru, zato tudi C sliši signal od B, ampak ker njemu ni namenjen, je tiho



A hoče poslati B, zato najprej pošlje RTS. B pošlje A CTS, ampak ta CTS vidi tudi C, zato istočasno C ne pošilja - je tiho.

# Dostop do medija: 802.11 WiFi

## Poznamo dve posebni situaciji

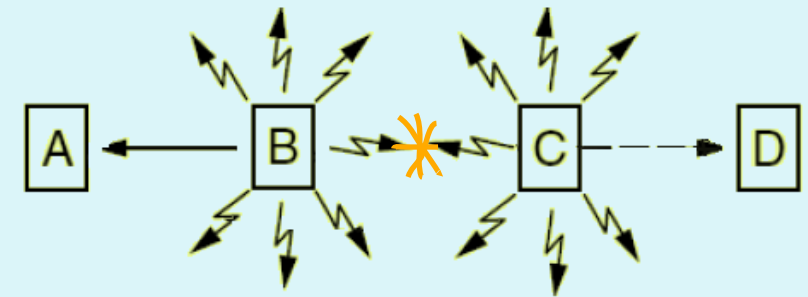
(exposed terminals)

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

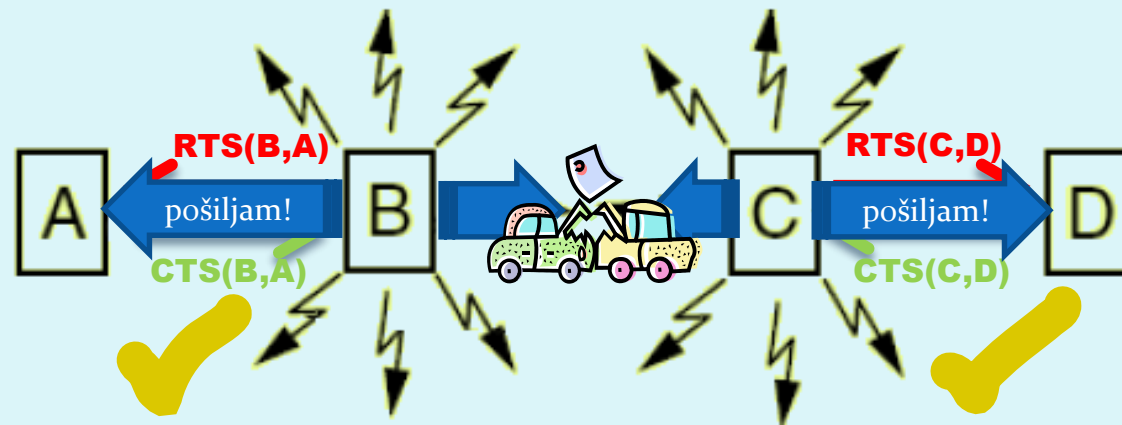
- terminala sta v dosegu (= sta izpostavljena)
- terminala ustvarjata trke (kolizije) **v vmesnem prostoru**
- vendarle pa ne ustvarjata trkov pri pošiljanju B  $\rightarrow$  A in C  $\rightarrow$  D

(gledamo zaporedja štirih postaj)

kolizija v vmesnem prostoru  
ni problematična



B pošilja A, C pošilja D in kolizij ni



prenosa B  $\rightarrow$  A in C  $\rightarrow$  D sta  
uspešna, v vmesnem prostoru  
pride do kolizije

# Vaja

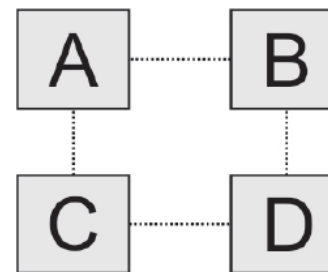
- notacija:  
"A -> B" ali "A pošilja B" pomeni: "Ali se lahko okvir uspešno prenese od pošiljatelja A do prejemnika B" (ne zahtevamo, da B nanj tudi odgovori ali da se vzpostavi daljša komunikacija)

- pozor, **popravek** strani 53 v učbeniku

skrita terminala (A in D)

izpostavljena terminala (A in C)

★ **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). S slike vidimo, da sta v dosegu vsakega izmed štirih vozlišč dve preostali vozlišči. Zastavimo si in odgovorimo na dve vprašanji:



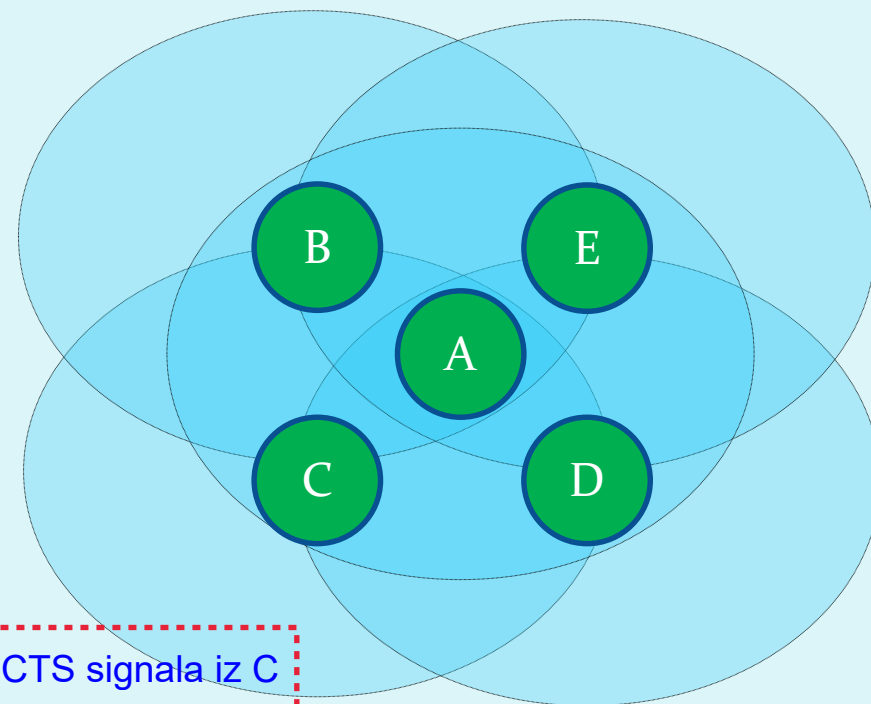
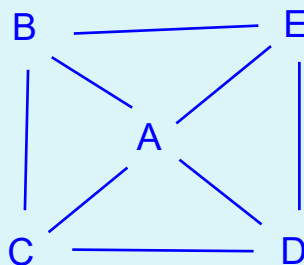
1. Ali lahko medtem, ko vozlišče A pošilja vozlišču B, tudi vozlišče D pošilja vozlišču C? *Odgovor je ne. Namreč, če vozlišče A pošlje RTS vozlišču B, bosta CTS vozlišča B slišala vozlišči A in D. D zato v času, ko A pošilja, ne bo oddajal.*
2. Ali lahko medtem, ko vozlišče A pošilja vozlišču B, tudi 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



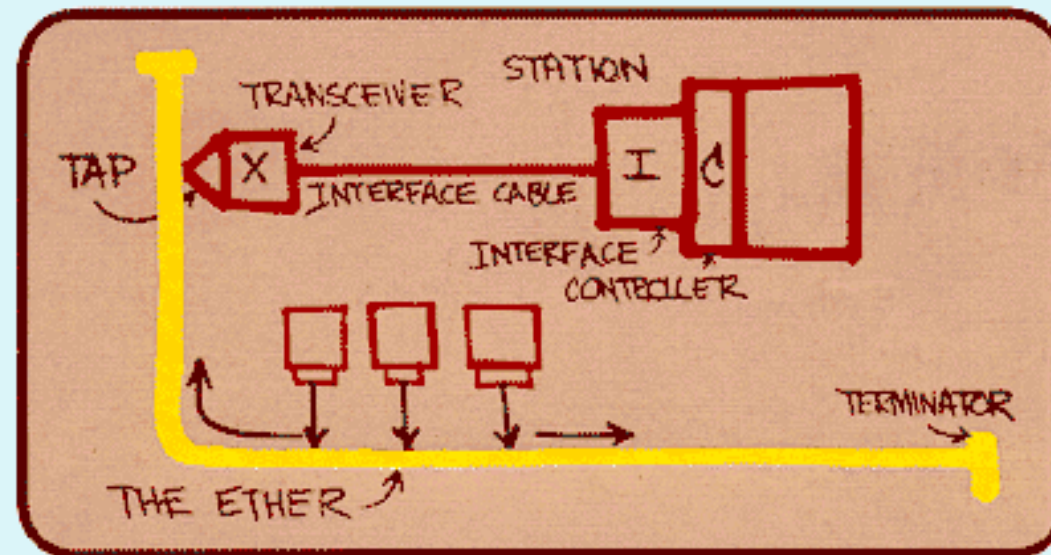
- vprašanja:

- kadar pošilja B -> C, ali lahko pošilja A -> D? ne, ker molči zaradi CTS signala iz C
- kadar pošilja B -> C, ali lahko E -> D? da izpostavljena terminala (B in E)
- kadar pošilja B -> A, ali drži, da ne more pošiljati nihče drug? da, ker so vsi ostali (C,E,D) v dosegu njegovega CTS
- kadar pošilja A -> B, ali lahko C -> A? ne, ker je v dosegu CTS signala od B
- kadar pošilja A -> B, ali lahko D pošilja komurkoli? (\*\*\*) ne, (D,E,B) so v postavitvi skritih terminalov

# Ethernet

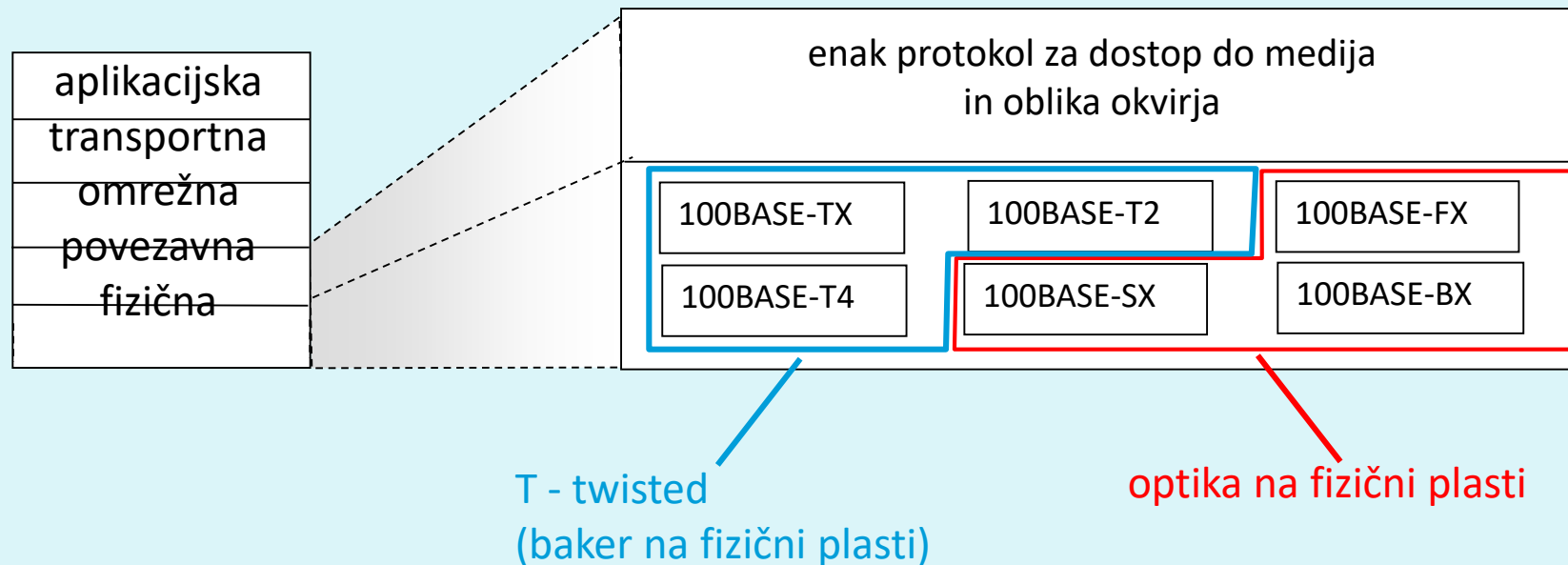
- razvoj v 70. - 90. letih na podlagi token ring, FDDI, ATM
- razlogi za uspeh:
  - hiter (hitrost do 100 Gbps),
  - poceni,
  - enostavnejši v primerjavi z drugimi
- skica koncepta snovalca Ethernet (Bob Metcalfe):

(topologija vodila na sliki)



# 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

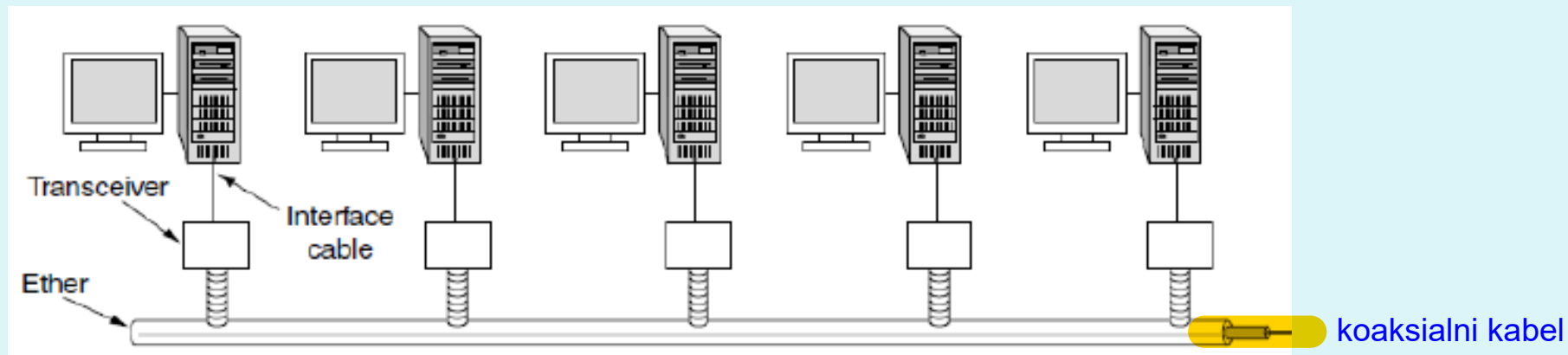


# Ethernet skozi čas

- zgodnji Ethernet

takrat je bilo treba za vsako povezavo na kabl presekati kabel in namestiti T-člen

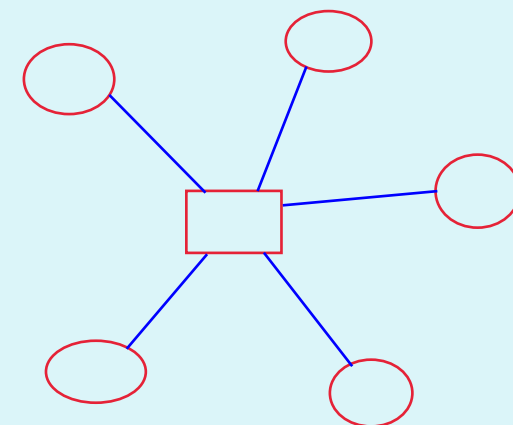
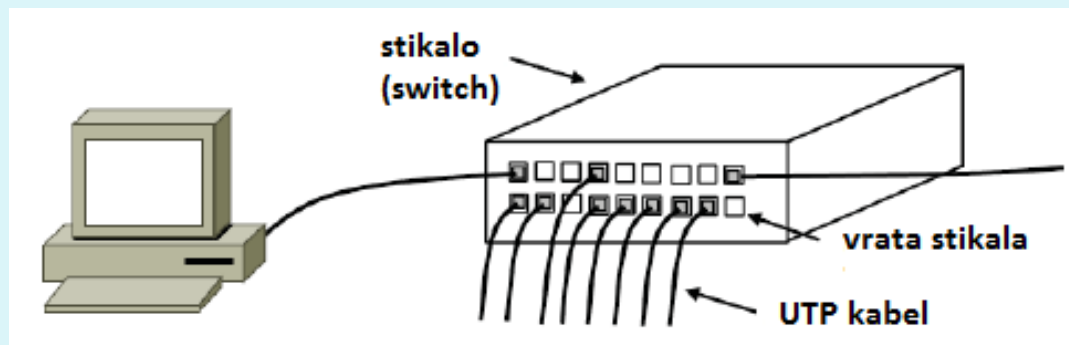
topologija vodila



SPF (single point of failure): kabel!

- sodobni Ethernet

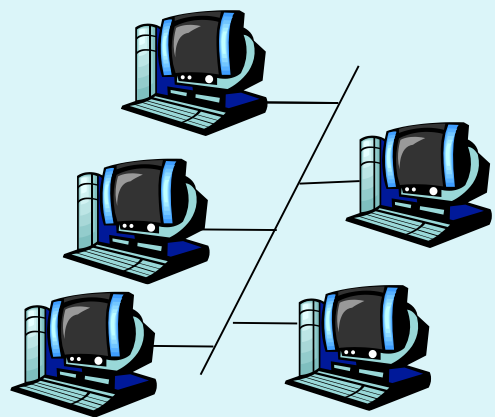
topologija zvezde



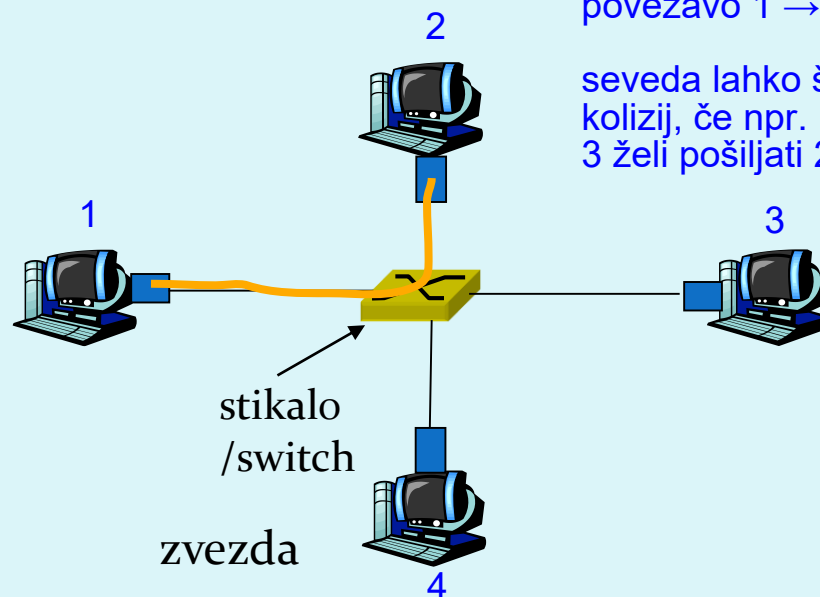
# Razvoj topologije Etherneta

- včasih: vodilo (bus)
  - vsi vmesniki v isti kolizijski domeni ko eden pošilja in kdorkoli drug spregovori, naredi kolizijo
  - najprej koaksialno vodilo (coax), nato topologija zvezda (naprava *hub* – razdelilec/zvezdišče)
- danes: zvezda (*stikalo* v centru) - okoli leta 2000
  - ločene kolizijske domene (Ethernet uporablja CSMA/CD)
  - možne različne različice Etherneta na vsakem kraku zvezde

primitivna naprava - signal samo razdeli, tj. vse povezane naprave ga dobijo (switch usmerja)



vodilo: koaksialni kabel

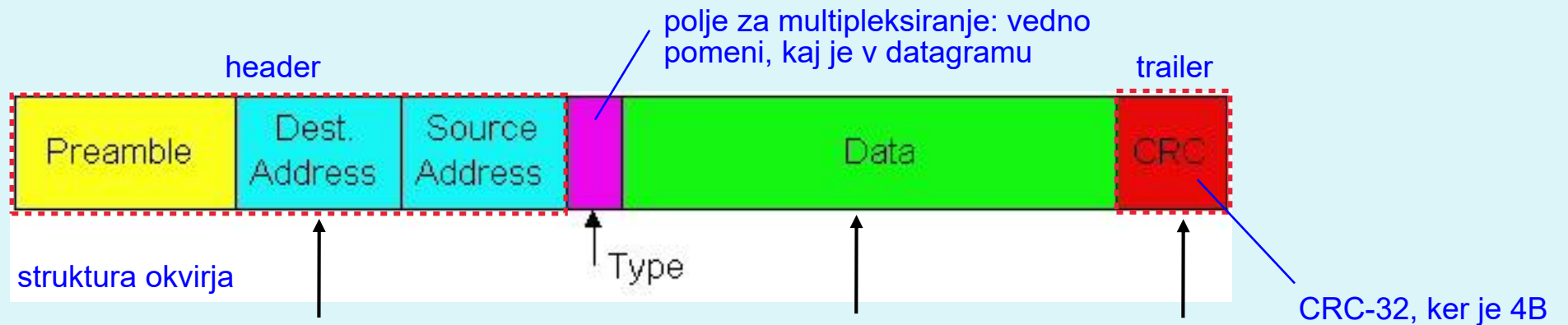


če 1 želi poslati 2 nekaj, je switch dovolj pameten, da vzpostavi samo povezavo 1 → 2

seveda lahko še zmeraj pride do kolizij, če npr. 1 pošilja 2, in hkrati 3 želi pošiljati 2 ipd.

# Okvir Ethernet

- Preambula: 7 x 10101010 in 1 x 10101011 to zaporedje bitov pomeni preambulo; dolžina 64 bitov (8 bajtov)
  - namenjena sinhronizaciji ur oddajnika in prejemnika (in hitrosti) zato da vesta, kako brati signal/podatke
  - znamenje: "pozor, sedaj prihajajo pomembni podatki"



♦ MTU je parameter, ki določa dolžino/velikost okvirja

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 ♦ = datagram/paket omrežne plasti

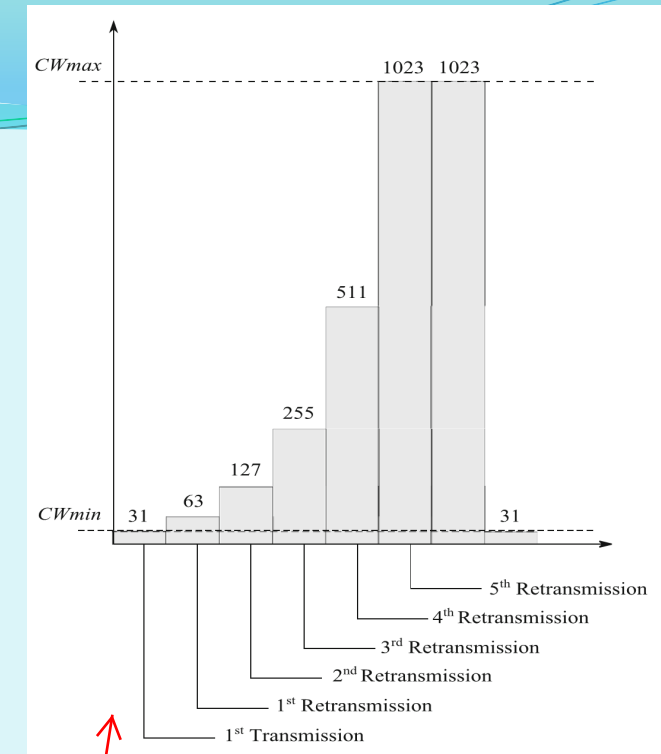
4B: Če ni ok, prejemnik okvir zavrže

zavrže, ker je popravljanje preveč časovno zahtevno

zgradba paketa omrežne plasti je odvisna od protokola: npr. IP, ARP, itd.

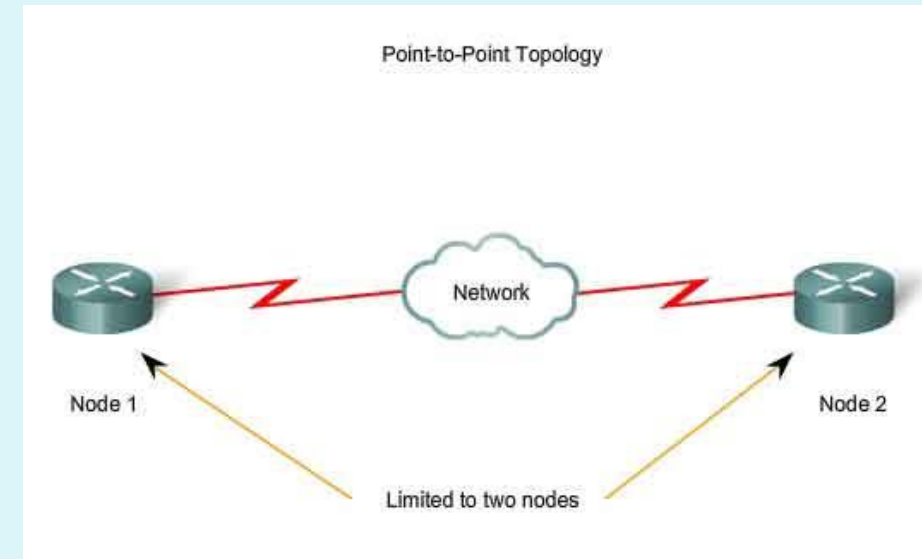
# Kaj ponuja Ethernet?

- **nepovezavna** storitev (ni rokovanja, vmesnik pošlje brez faze vzpostavljanja)
- **nezanesljiva** storitev
  - kontrola pravilnosti se izvaja s CRC
  - potrjevanje in ponovno pošiljanje se ne uporabljata!
  - omrežna in transportna plast skrbita za vrstni red in prenos vseh podatkov  
napake se odpravljajo na višjih plasteh
- **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 možnosti, 50% uspešnost
    2. ob drugi koliziji pošlje po čakanju 0-3 okvirje 4 možnosti, 75% uspešnost
    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 zavržejo)
  - **učinkovitost** rabe medija 85 – 100 %!



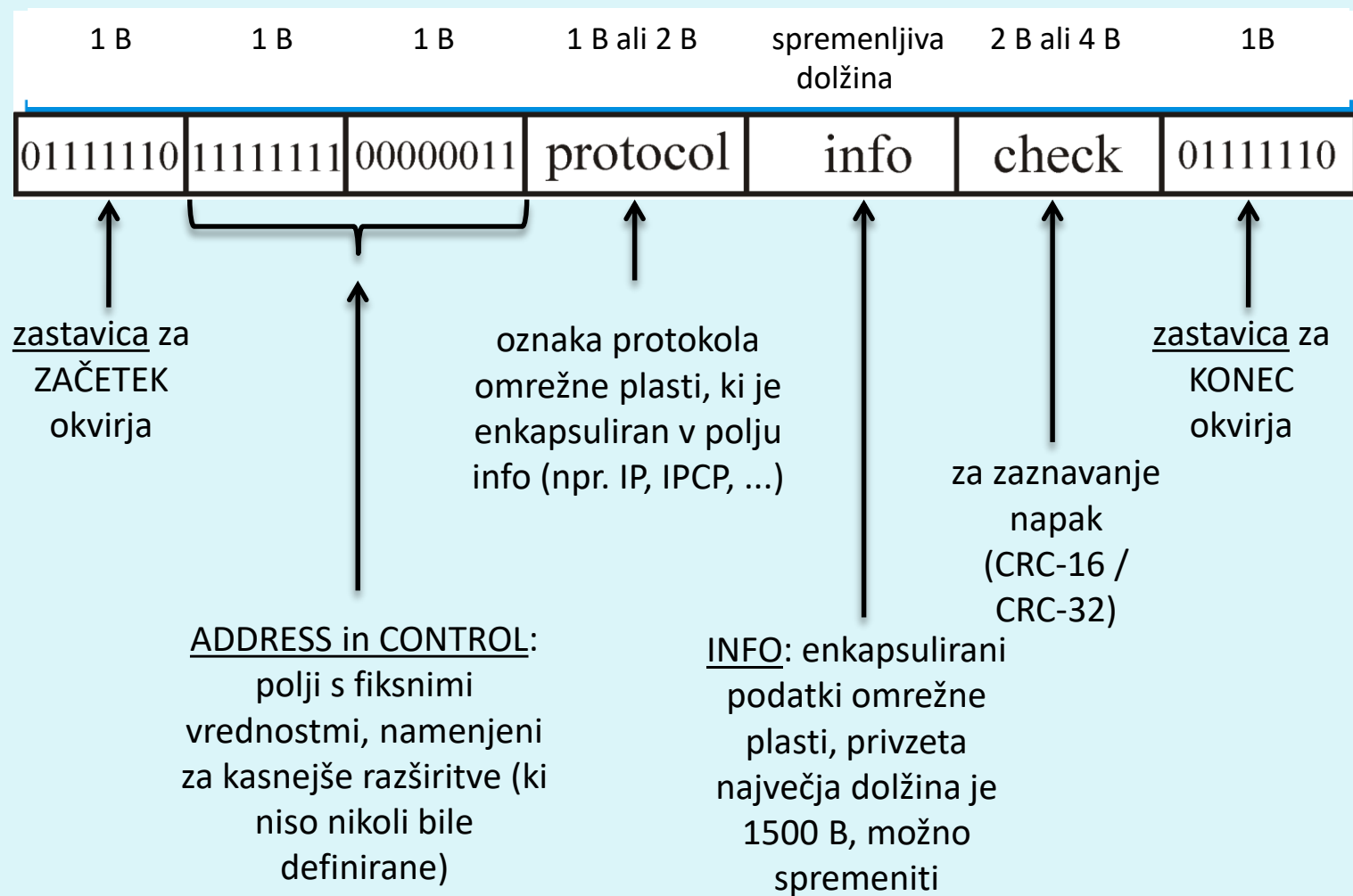
# Protokol PPP

- drugi povezavni protokol, pogledjmo 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



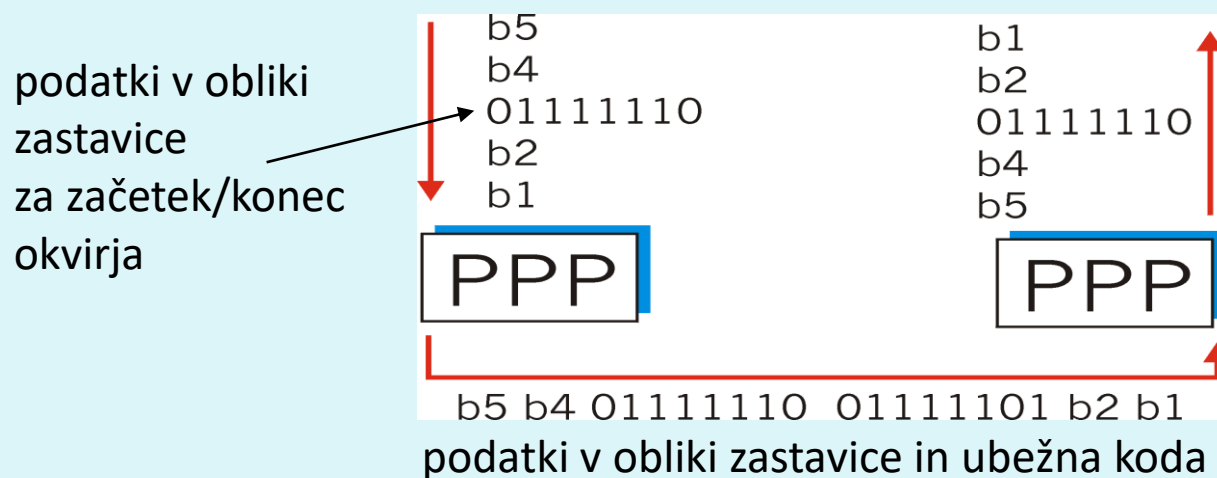


# Okvir PPP



# 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  $2^{48}$  naslovov, prva polovica je ID proizvajalca, druga polovica ID adapterja (podjetje zakupuje pakete po  $2^{24}$  naslovov)
  - 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

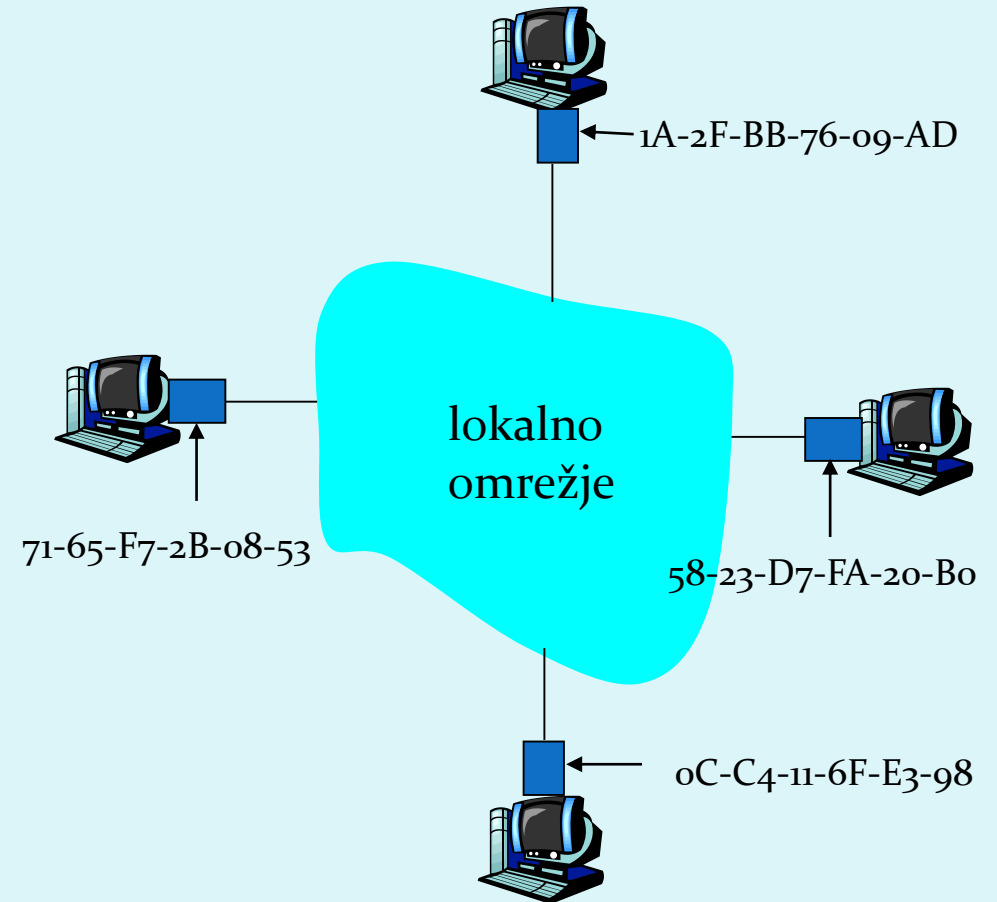
Ethernet adapter Local Area Connection:

    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 naslovov 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)

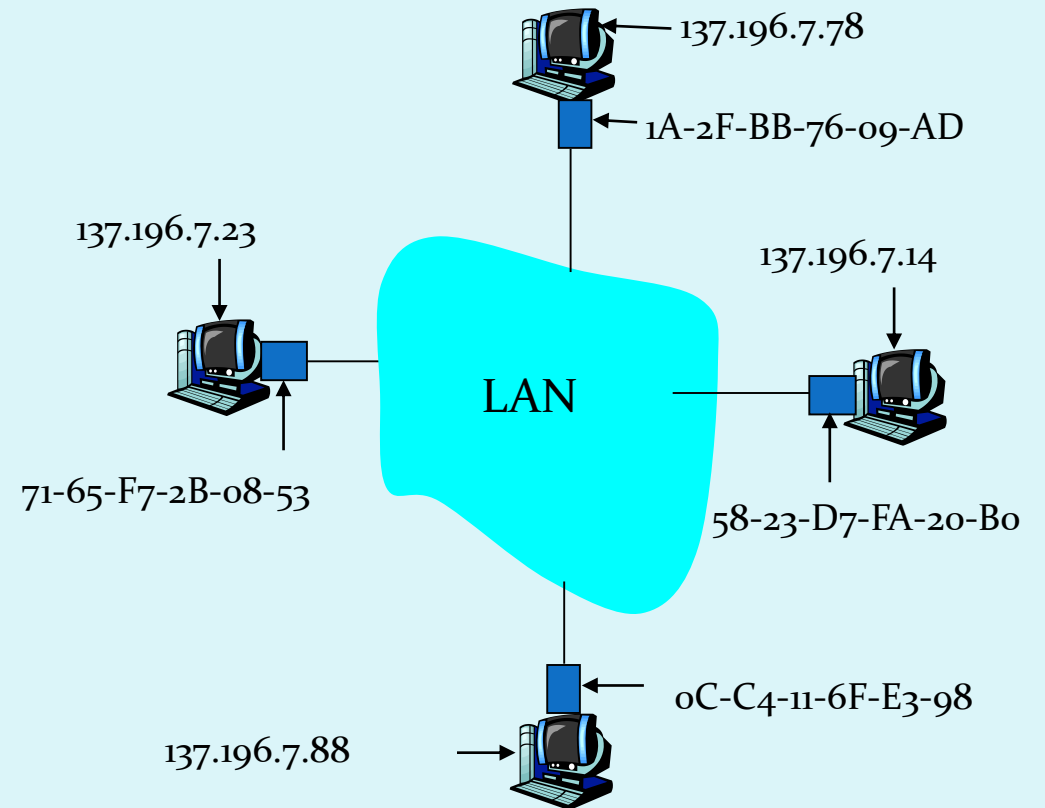


# Vendar pa ...

- (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 preda 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*!

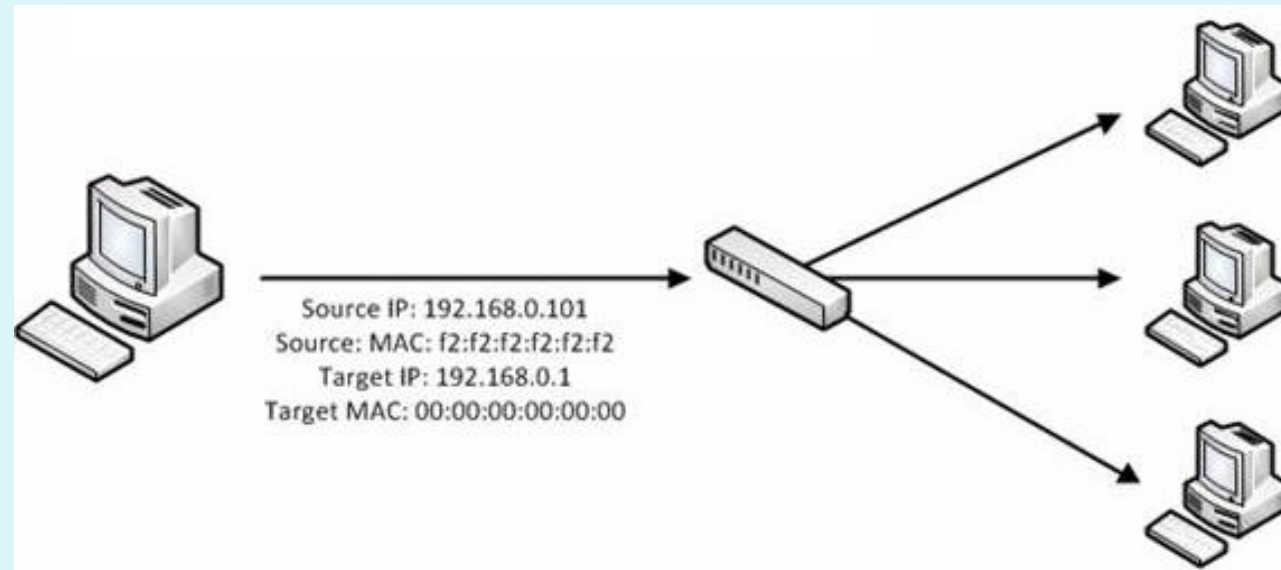
# ARP: Address Resolution Protocol

- 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 celem internetu
- Kako se tabela ARP napolni?
  - uporaba paketov ARP query



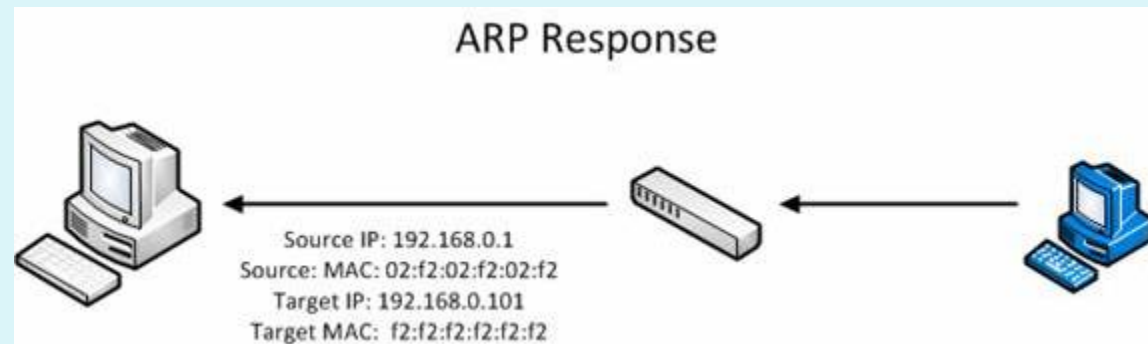
# ARP: nastanek tabele ARP

- 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?"





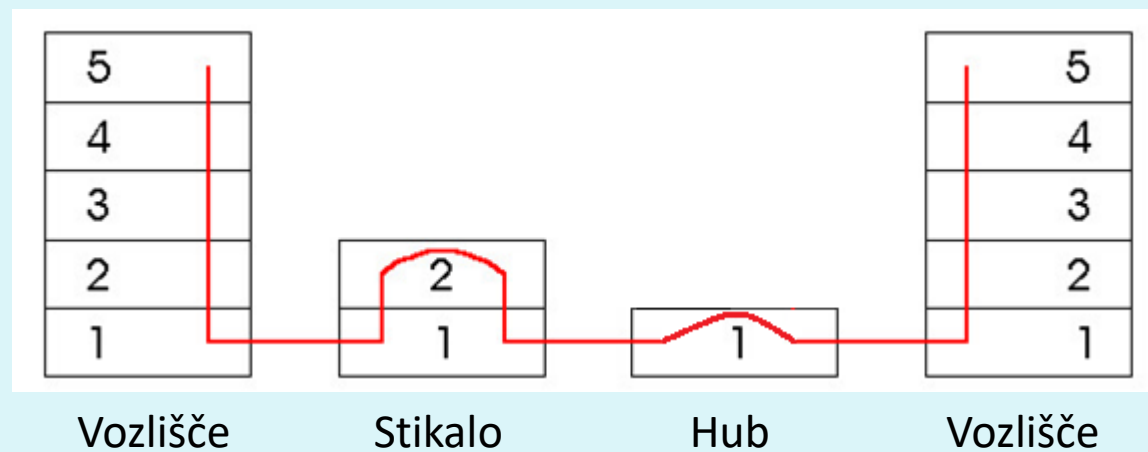
# Aktivna oprema (naprave)

- za povezovanje omrežne opreme se uporabljajo:
  - **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, poplavljanje in filtriranje**



# Primerjava

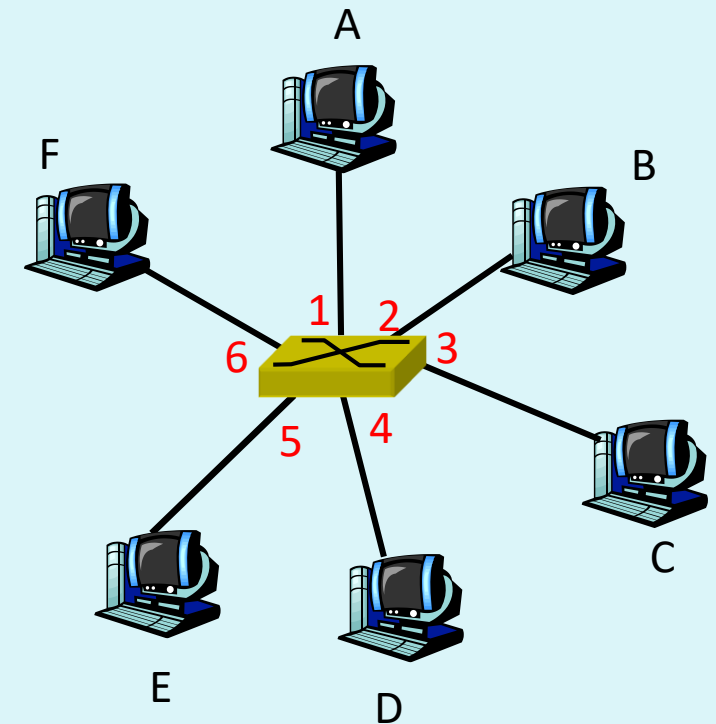
	Hub	Stikalo
Izolacija prometa	Ne	Da
Potrebna konfiguracija?	Ne	Ne



# 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



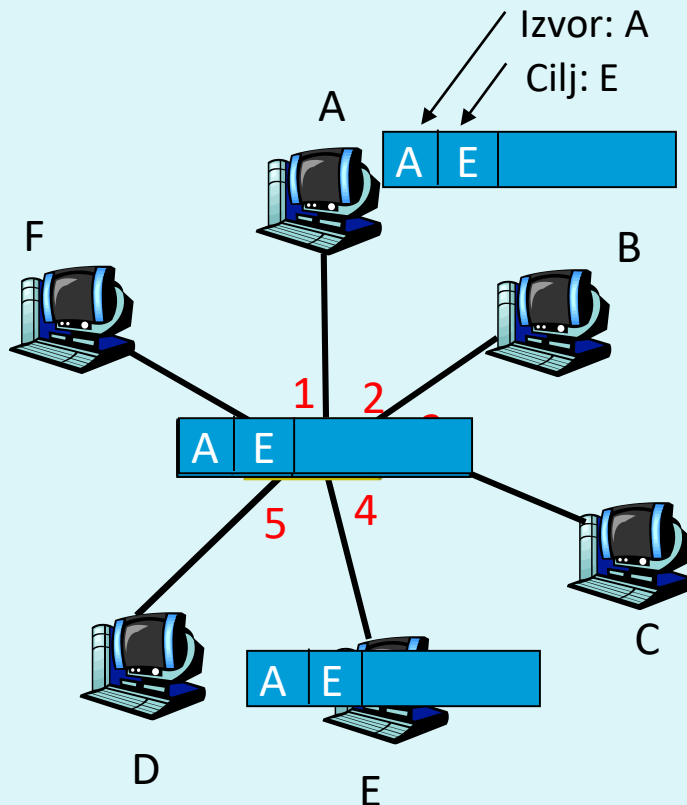
stikalo s 6 vmesniki

# 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, zavržemo ga)

# Prvi sprejem okvirja

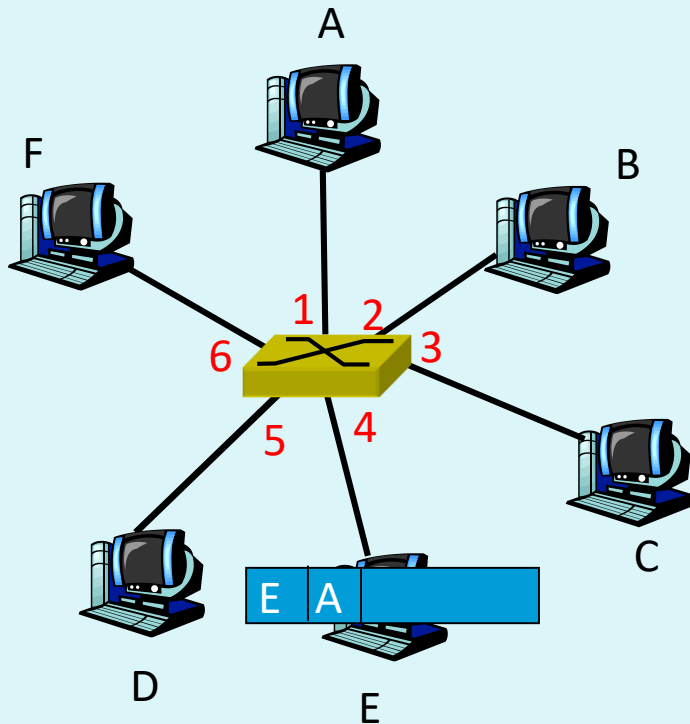
- stikalo **shrani lokacijo** pošiljatelja A
- ker stikalo ne ve, kje je cilj, **poplavi** (flood) okvir na vsa vrata



naslov MAC	vrata	TTL
A	1	60

# Pošiljanje znanemu prejemniku

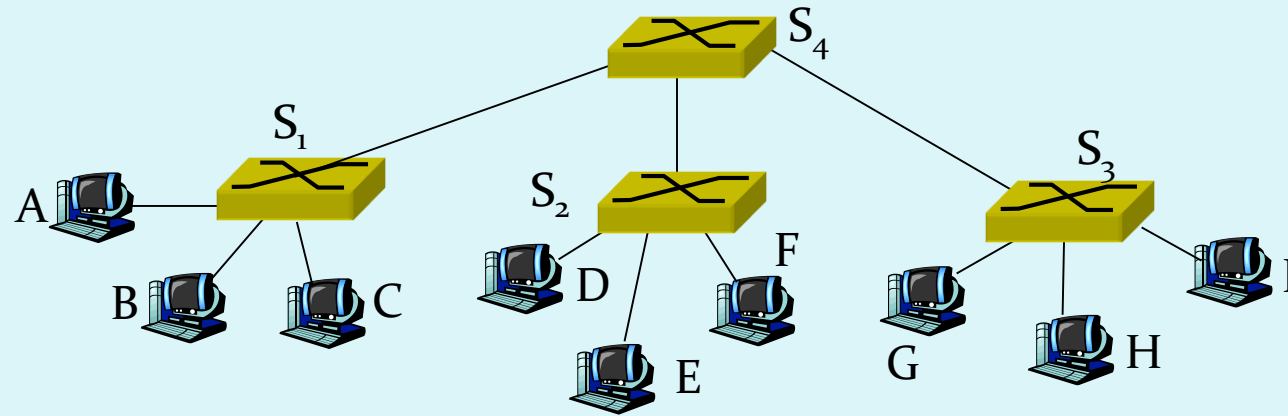
- stikalo prejme okvir za **znanega prejemnika (A)**
- ker stikalo ve, kje je cilj, **posreduje** okvir na prava vrata (1)



naslov MAC	vrata	TTL
A	1	60
E	4	60

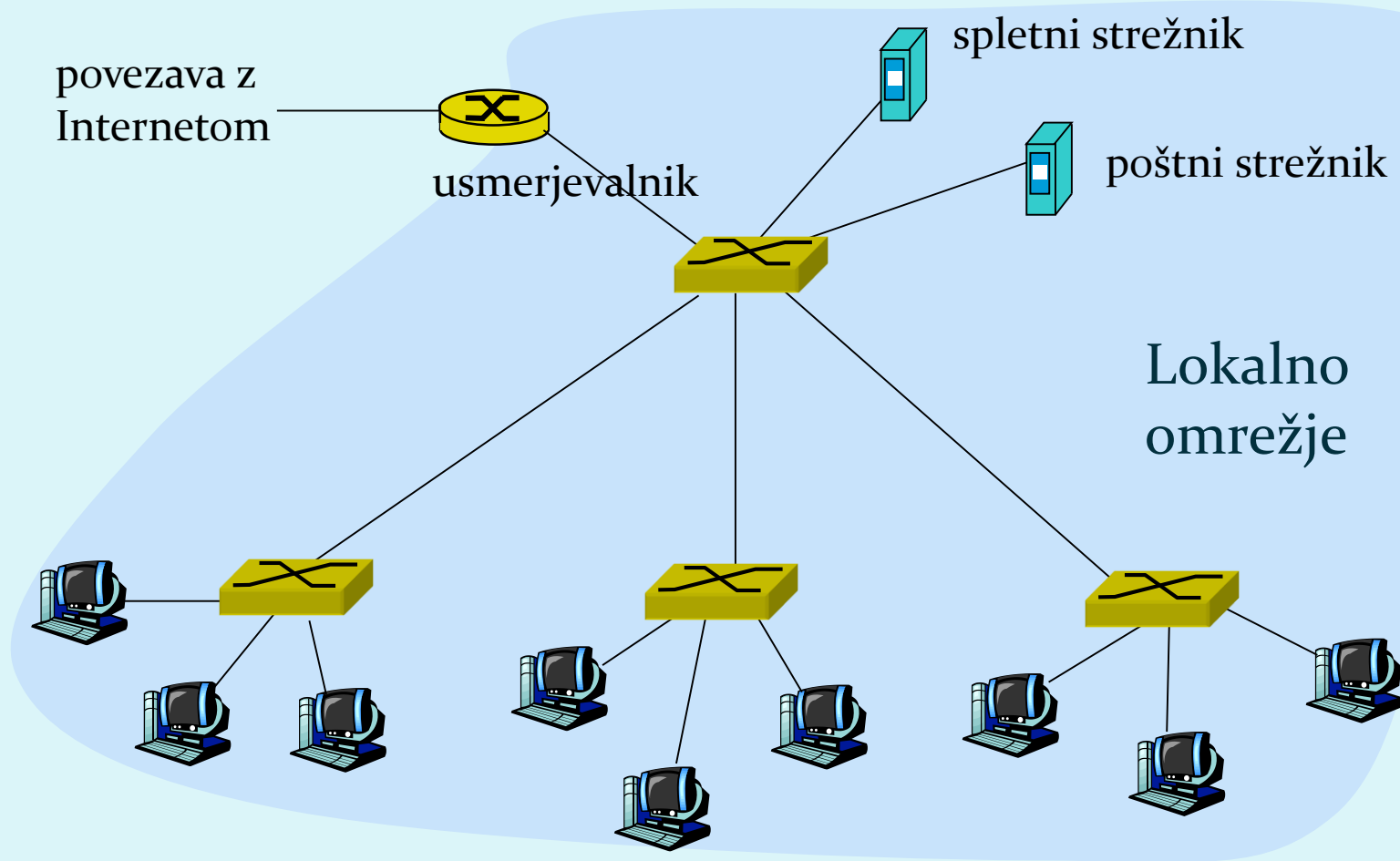
# Povezovanje stikal

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



- **FILTRIRANJE:** stikalo S<sub>4</sub> bo zavrglo okvir, ki ga npr. A pošlje C, ker je na istih vratih (če ima stikalo S<sub>4</sub> v stikalni tabeli zapis za C)

# Primer uporabe stikal

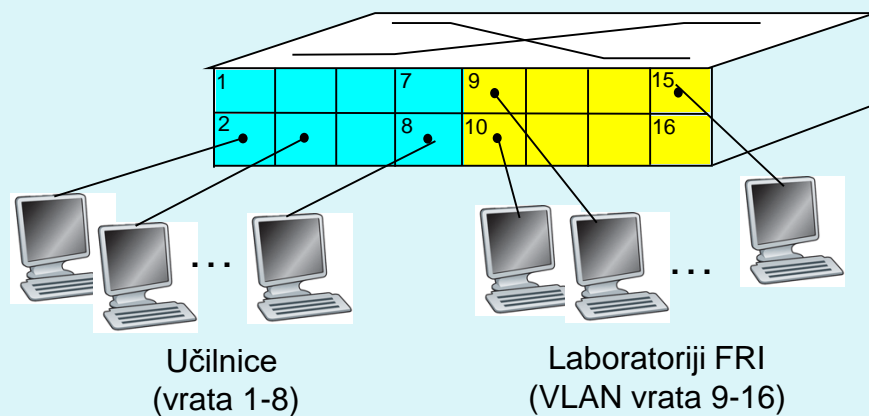




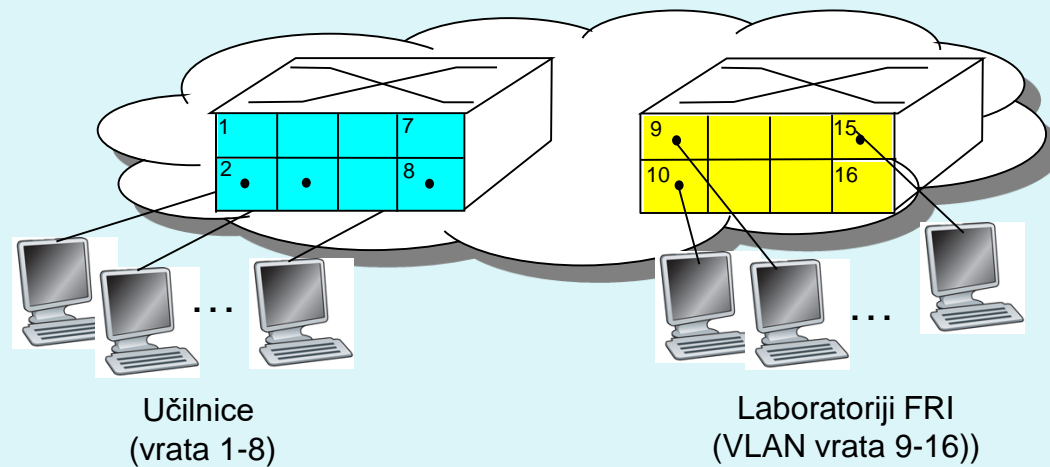
# Virtualna lokalna omrežja (VLAN)

- 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



# Naslednjič gremo naprej!

- omrežna plast!

