

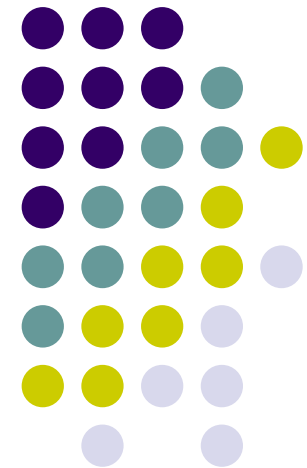
Lekcija 5:

Podatkovni sloj OSI modela

Prof.dr.sc. Jasmin Velagić
Elektrotehnički fakultet Sarajevo

Kolegij: Distribuirani sistemi

2012/2013



Sadržaj poglavlja:

- ✚ Podatkovni sloj OSI modela
- ✚ Podsloj okvira (Frame sub-layer)
 - Kreiranje okvira
 - Otkrivanje grešaka
 - Ispravljanje grešaka
- ✚ Podsloj pristupa mediju (Medium Access Control, MAC)
- ✚ Podsloj logičkog sloja (Logical Link Control)



5. PODATKOVNI SLOJ OSI MODELA

Funkcije podatkovnog sloja

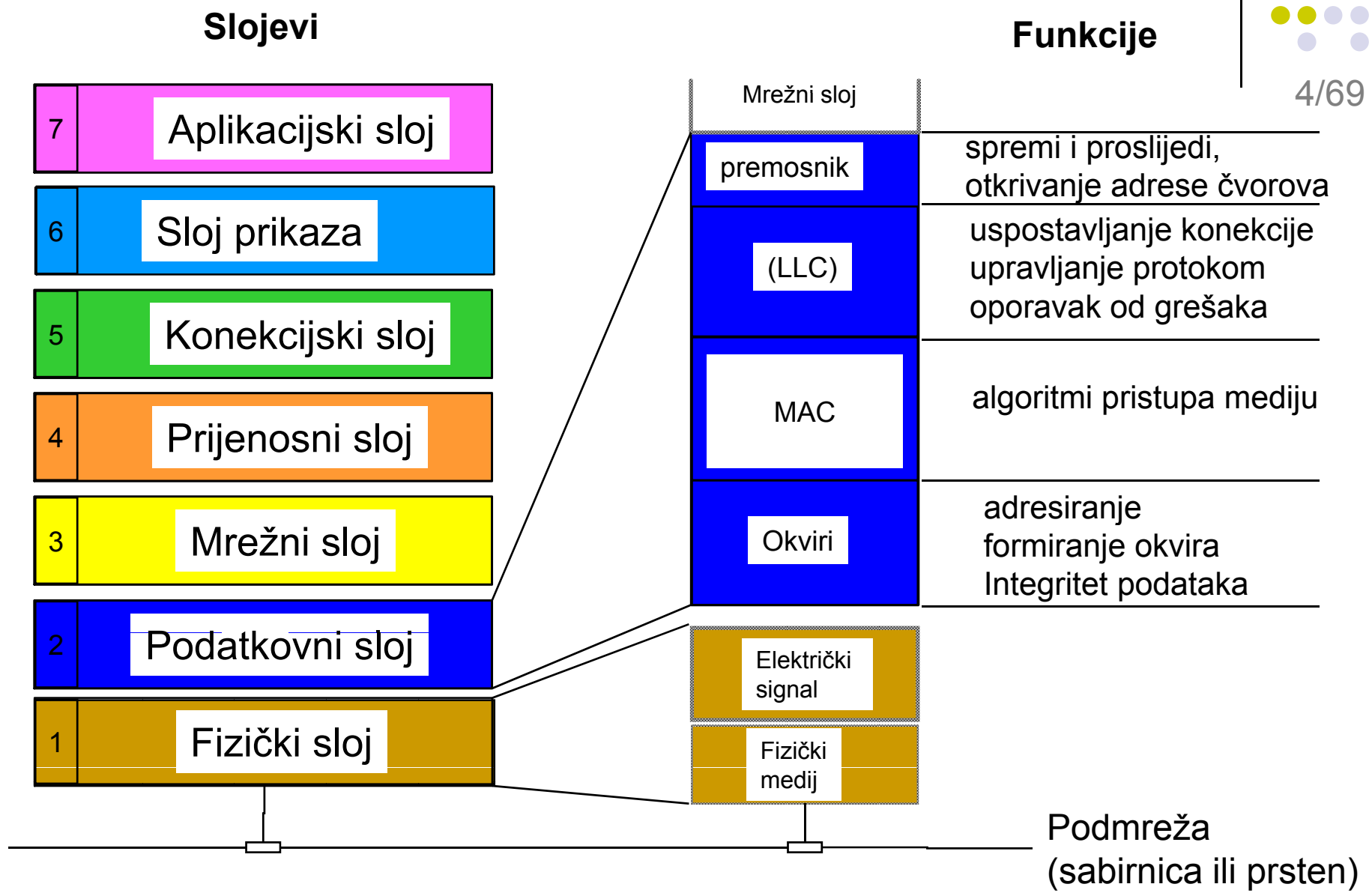
- Protokoli podatkovnog sloja osiguravaju komunikaciju unutar iste pod mreže.
- **Pod mreža**: ista brzina prijenosa, isti način pristupa mediju, prijenosni mediji mogu biti različiti (obnavljači signala).
- Zadaci (funkcije) podatkovnog sloja:
 - **Integritet podataka (Data Integrity).**
 - **Pristup prijenosnom mediju (Medium Access Control, MAC).**
 - **Upravljanje logičkom vezom (Logical Link Control).**
 - **Upravljanje podatkovnim slojem (Link Layer Management).**

Podatkovni sloj mora:

- **Osigurati dobro definirano sučelje prema mrežnom sloju (pružanje usluga mrežnom sloju).**
- **Definirati način na koji se bitovi sa fizičkog sloja grupiraju u okvire.**
- **Riješiti problem grešaka pri prijenosu.**
- **Regulirati (upravljati) protok okvira.**



5.1. Podatkovni sloj unutar OSI modela

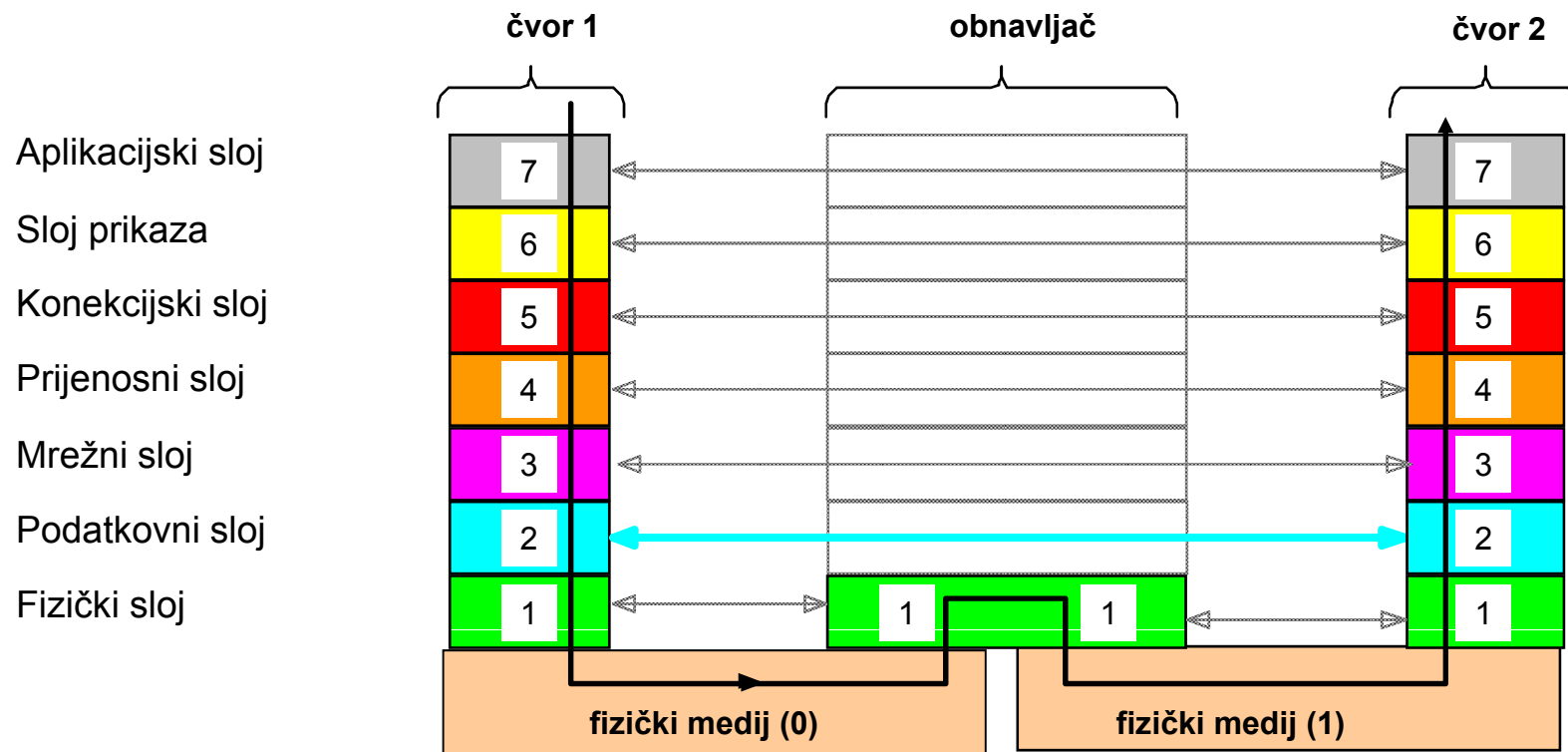


Podatkovni sloj unutar OSI modela

OSI modeli povezani obnavljačem



5/69

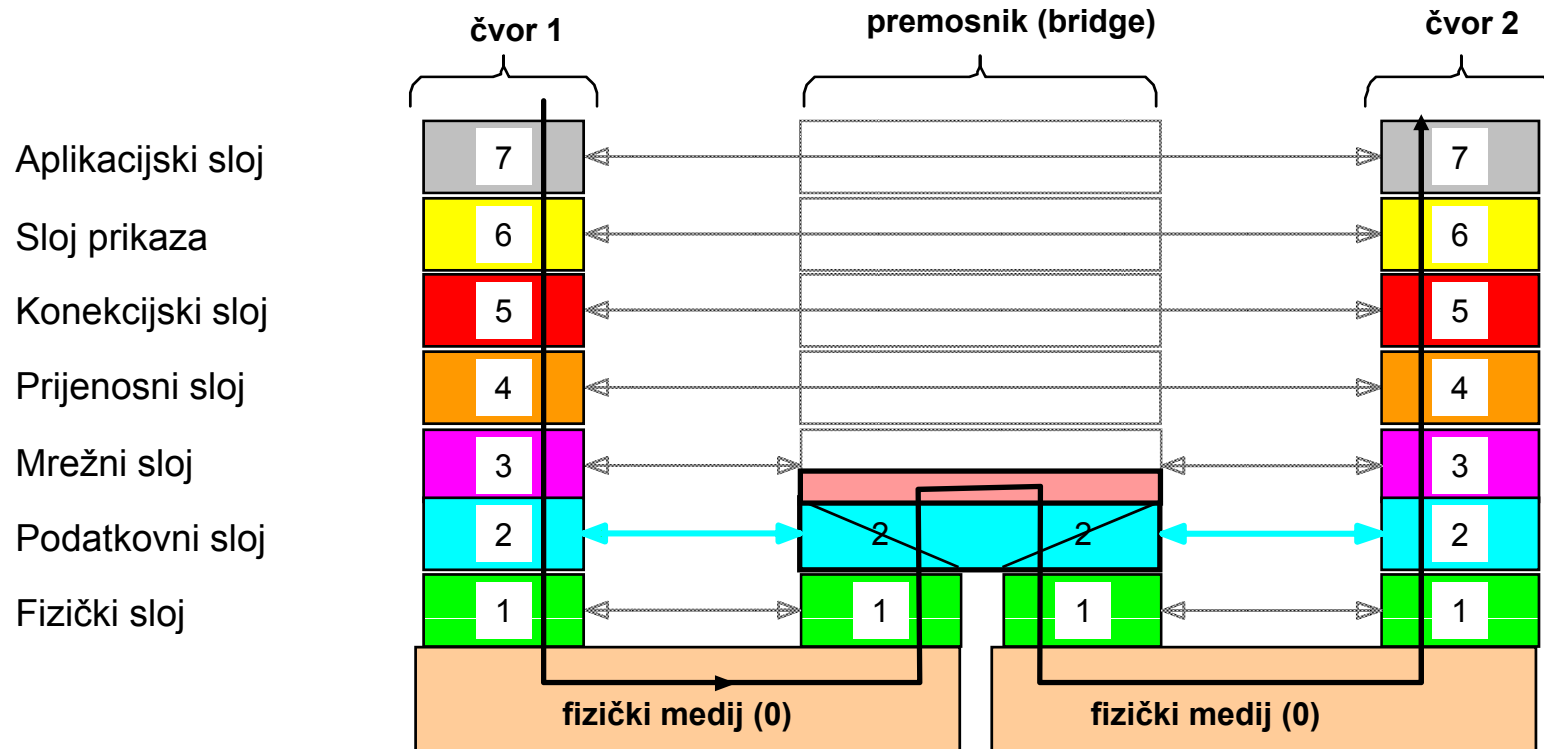


Dva segmenta na svakoj strani obnavljača formiraju pojedinačnu podmrežu sa:

- **istim brzinama** (medij, modulacija se može razlikovati),
- **jednakim formatima okvira** (izuzev rubnih efekata),
- **istim medijima pristupa**,
- **istim adresnim područjem** (transparentno na obje strane obnavljača).

Podatkovni sloj unutar OSI modela

OSI modeli povezani premosnikom



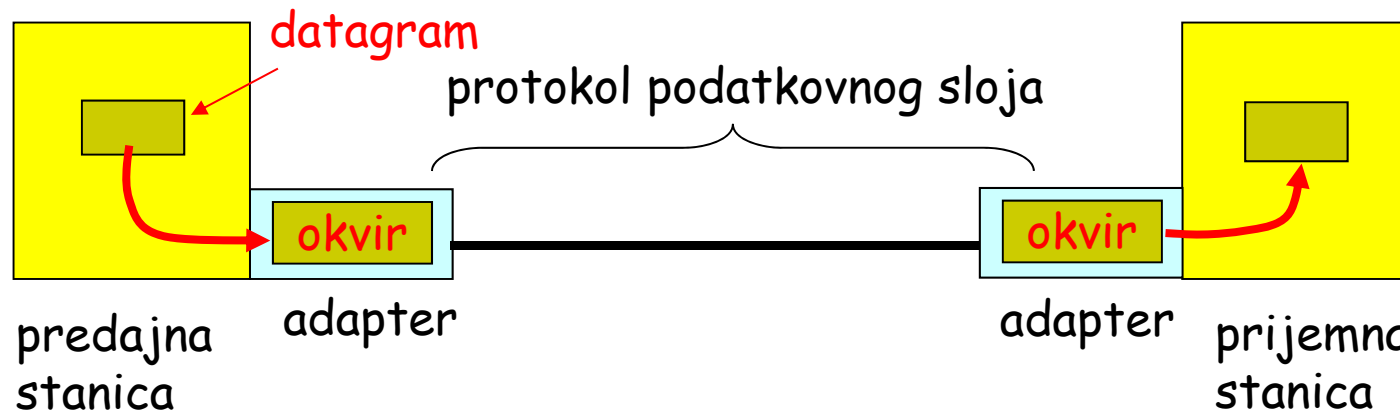
Podmreže s obje strane premosnika imaju:

- **jednake formate okvira** (osim zaglavlja),
- **isto adresno područje** (različite adrese čvorova sa obje strane),
- **jednake protokole podatkovnog sloja.**

Premosnici filtriraju okvire na temelju njihovih adresa u podatkovnom sloju.

Podatkovni sloj unutar OSI modela

Povezivanje podatkovnih slojeva – komunikacija adaptera

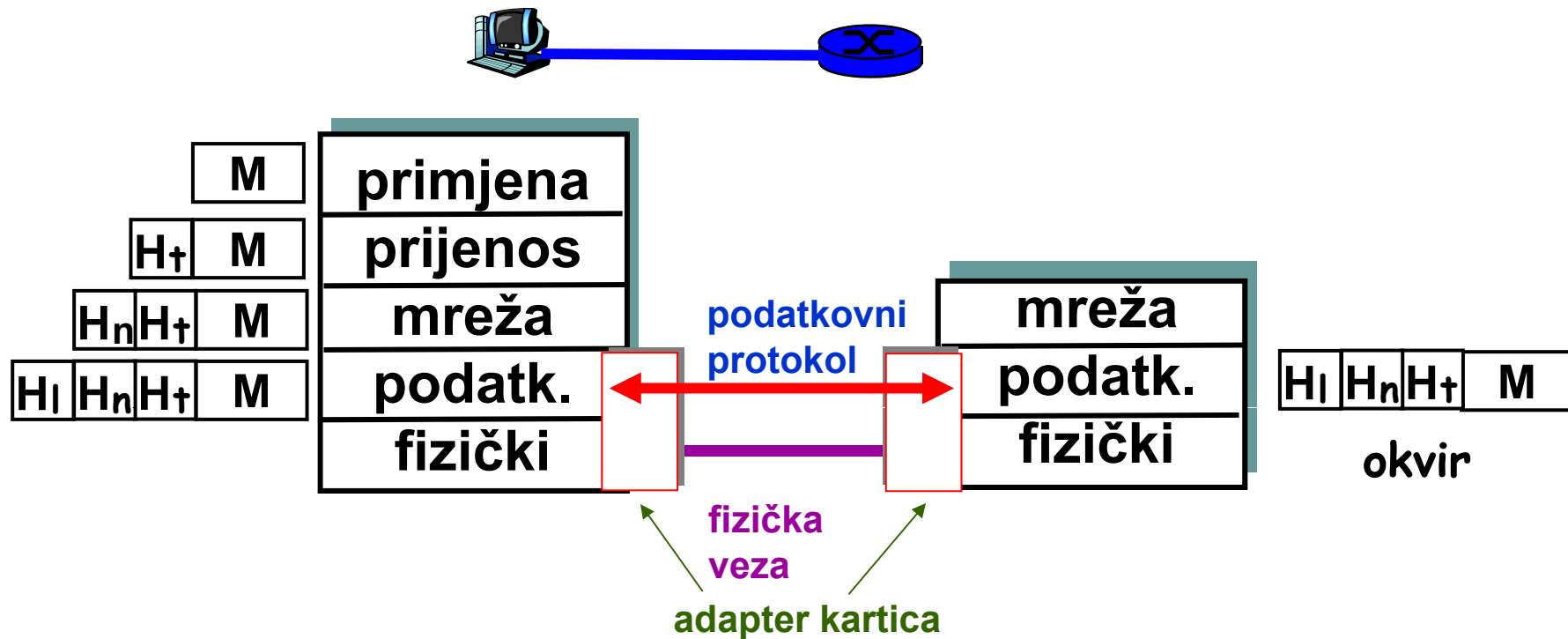


- podatkovni sloj implementiran u “adapteru” (aka NIC)
 - Ethernet, PCMCIA i 802.11 kartice
- **predajna strana:**
 - sažimanje datagrama u okvir
 - dodavanje bitova za provjeru greške, rdt, upravljanje protokom, itd.
- **prijemna strana**
 - oporavljanje od greški, rdt, upravljanje protokom, itd.
 - ekstrakcija datagrama, prolazak do prijemne stanice
- adapter je poluautonoman
- podatkovni i fizički slojevi

Podatkovni sloj unutar OSI modela

Povezivanje podatkovnih slojeva – komunikacija adaptera

- ❑ Adapter sadrži RAM, DSP čipove, host bus sučelje, sučelje povezivanja.
- ❑ Na ovaj način se povezuju stanica i router, router i router, router i stanica.



HDLC protokol

HDLC kao primjer protokola podatkovnog sloja



Standard HDLC (High-level Data Link Control)

Funkcija Ostvarivanje prijenosa između uređaja pod mreže.

Objekti Strukture okvira u skladu sa ISO 3309.

Zastavica	Adresa	Kontrola	Podaci	CRC	Zastavica
01111110	8 bita	8 bita	(n · 8)	16 bita	01111110

Provjera integriteta 16-bit CRC (Cyclic Redundancy Check).

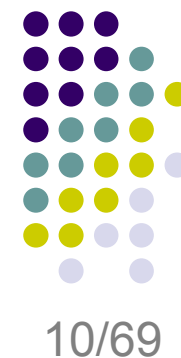
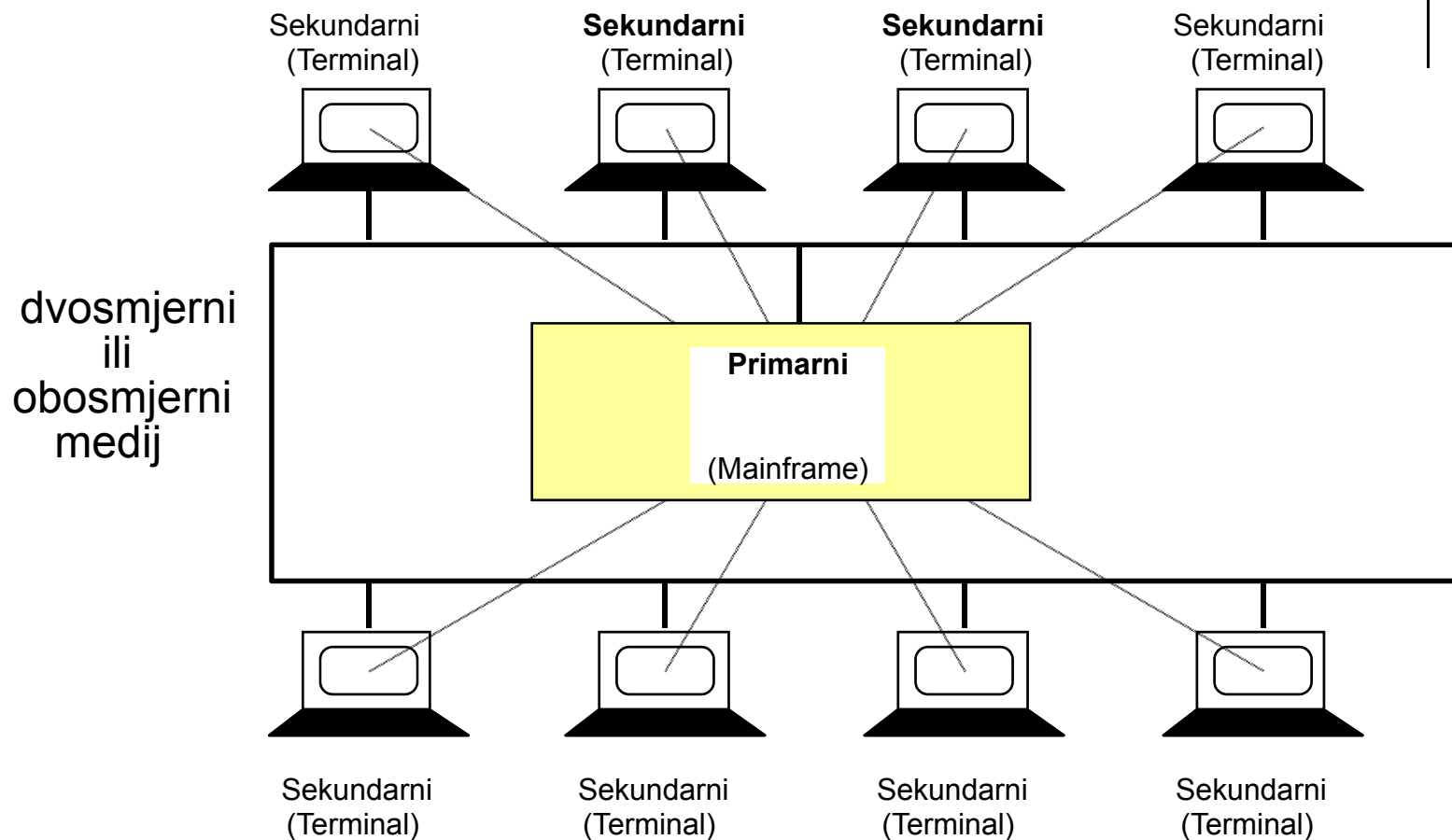
Pristup mediju Master/Slave, (sa mogućom inicijativom od strane slave-a).

Oporavak greške Potvrda sigurnosti i ponavljanje.

Upravljanje protokom 7-okvirni (127 okvira) kreditni sistem.

HDLC protokol

HDLC topologija



Primarni (Master) računar je povezan sa sekundarnim računarima (Slave-a) preko multidrop sabirnice (npr. RS 485) ili preko tačka-tačka sabirnice (linija).

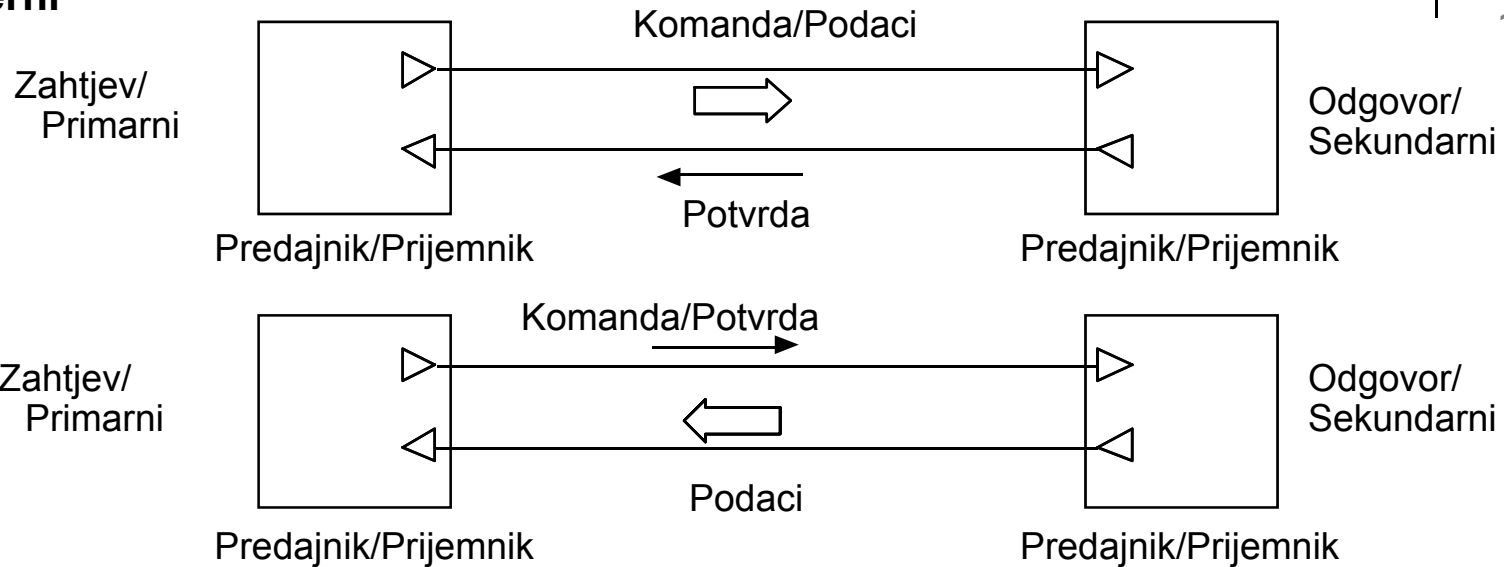
HDLC fizički je sabirnica, ali logički to je zvjezdasta mreža.

HDLC protokol

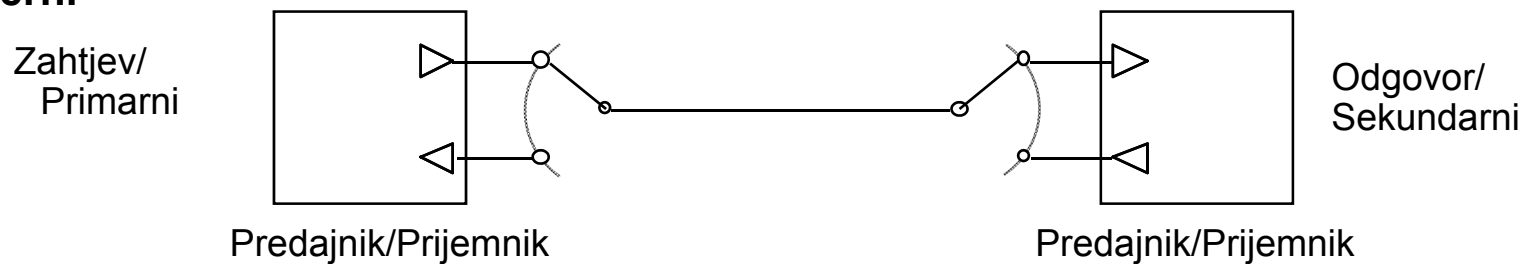
HDLC – dvosmjerne i obosmjerne operacije



Dvosmjerni

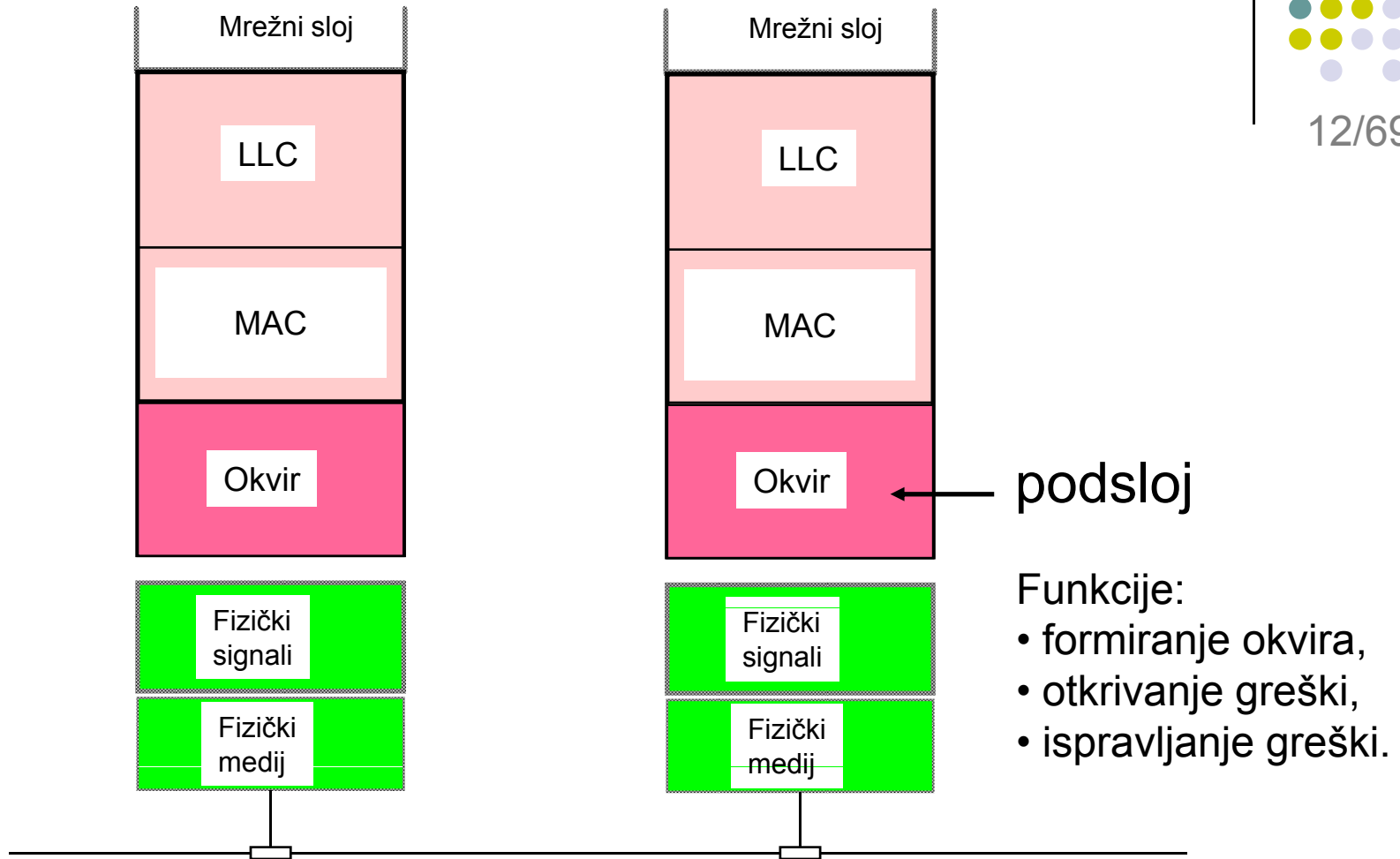


Obosmjerni



Primarni prebacuje sekundarnog u mod slanja, sekundarni šalje dok ne dobije natrag kontrolu.

5.2. Podsloj okvira



Podsloj okvira zadužen je za ispravan format i sadržaj okvira, neovisno o mediju i/ili brzini prijenosa.

5.2.1. Formiranje okvira

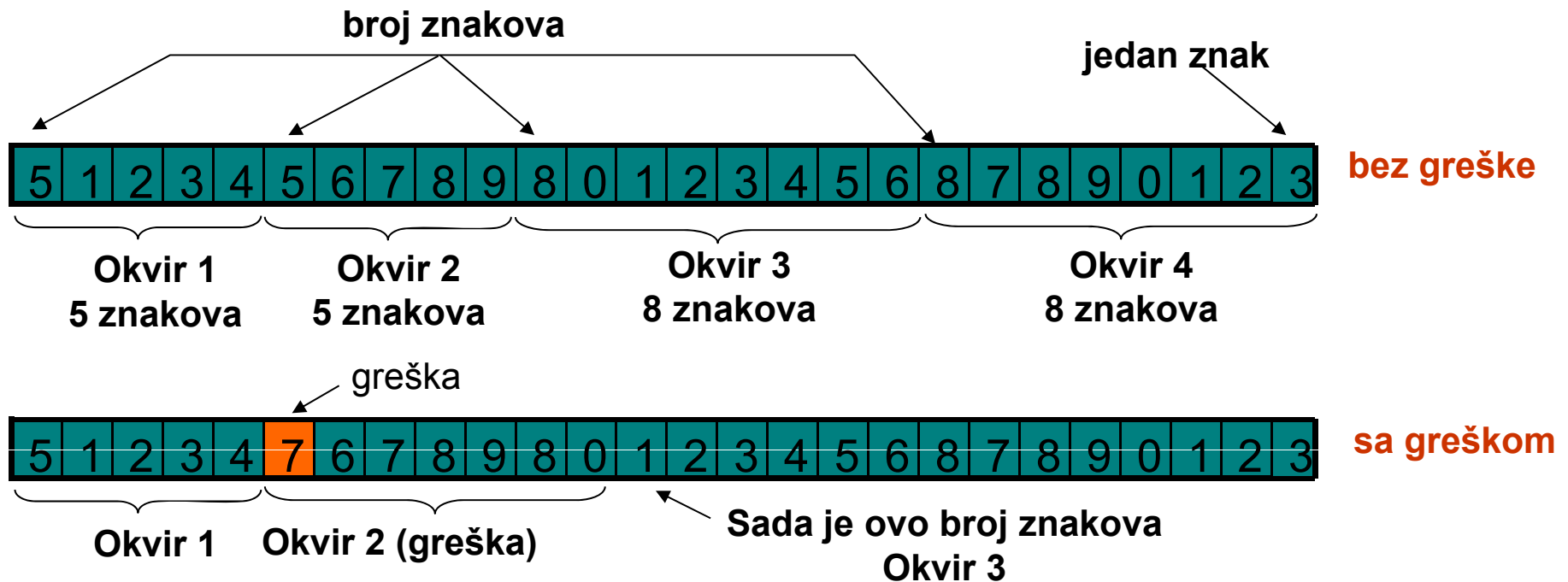
- Pretvara niz bitova fizičkog sloja u diskretne jedinice (okvire) i računa kontrolni niz bitova za svaki okvir.
- Na određenoj strani kontrolni niz bitova (polje za provjeru) se ponovo računa i ako se ne poklapa s vrijednošću izračunatoj na predajnoj strani znači da je došlo do greške.
- Problem detekcije početka i kraja okvira od strane prijemnika – problem sinhronizacije.
- Ovaj problem se može riješiti na sljedeće načine:
 - Brojanje znakova (Character Count),
 - Umetanje bitova (Bit Stuffing),
 - Umetanje znakova (Character Stuffing),
 - Narušavanje kodiranja (Encoding Violations) – samo za mreže sa kodiranjem u fizičkom sloju.



Formiranje okvira

Brojanje znakova

- **U prvom polju okvira upisana je njegova duljina.** Na temelju ovog polja prijemnik zna duljinu okvira i može odrediti gdje je njegov kraj.
- **Nedostatak je da prijemnik gubi sinhronizaciju kada su bitovi izmijenjeni tokom slanja.** Npr. ako se prilikom slanja promijeni broj u prvom polju, prijemnik će imati pogrešnu informaciju o duljini okvira.
- Iako će “provjerena suma” (checksum) otkriti da je okvir neispravno prenesen, prijemniku će biti teško ponovo se sinhronizirati. Zbog toga se ova tehnika više ne koristi.





Formiranje okvira

Umetanje bitova (Bit Stuffing)

- Bitno orijentirana metoda, dopušta da okviri imaju proizvoljan broj bitova, te kodiranje znakova sa proizvoljnim brojem bitova.
- Na svaki okvir se dodaje niz bitova 01111110 koji se zovu **rezervirani bitovi** i služe za označavanje početka, odnosno kraja okvira. Ovim nizom se razdvajaju dva susjedna okvira, tako da okvir sadrži bitove između ova dva niza.
- **Problem nastaje kada se rezervirani bitovi nađu unutar okvira.**
- **Rješenje: Umetanje bitova.** Predajnik nakon svakog niza od pet 1 ubacuje 0. Kad prijemna strana vidi pet jedinica iza kojih slijedi 0 ona je automatski izbacuje.

a) izvorni podaci

a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

b) poslani podaci

b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

c) primljeni podaci

umetnuti bitovi (nule)

c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

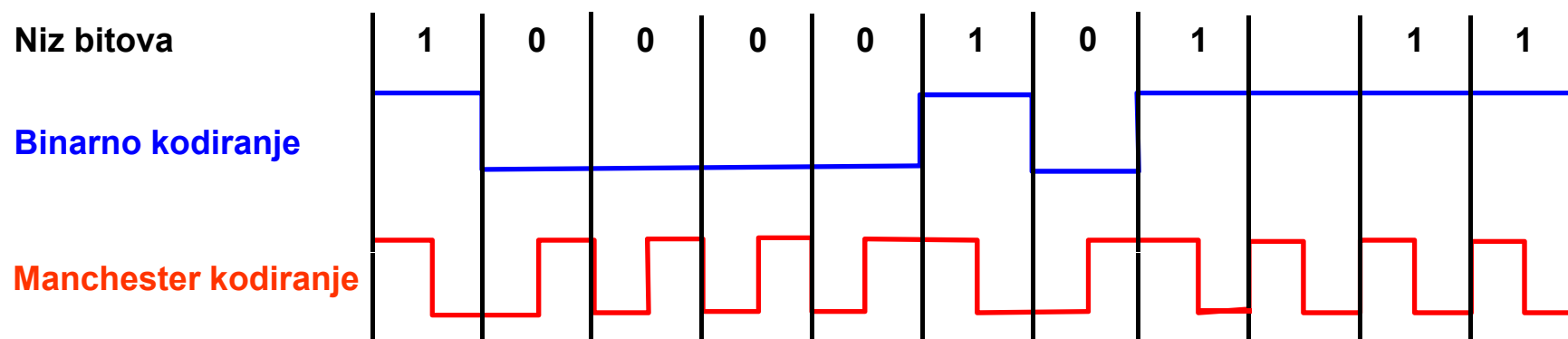
Nedostatak: Smanjena propusna moć medija zbog povećanja broja bitova. Pogodna samo za mreže kod kojih na fizičkom sloju postoji redundancija.



Formiranje okvira

Narušavanje kodiranja (Encoding Violations)

- Signali su kodirani u fizičkom sloju, npr. Manchester kodiranje: bit “1” je predstavljen parom *visoko-nisko*, a bit “0” parom *nisko-visoko*. Svaki bit ima prijelaz u sredini, što olakšava predajniku prepoznavanje bitova. Početak/kraj okvira mogu se predstaviti sa dva signala *nisko-nisko* ili *visoko-visoko*.
- Dvostruko smanjenje propusne moći.
- LAN-ov IEEE 802.4 standard koristi ovaj postupak.
- LAN IEEE 802.3 (Ethernet) koristi i polje sa oznakom duljine okvira i posebne nizove bitova za označavanje početka i kraja okvira.



5.2.2. Otkrivanje i ispravljanje grešaka

Osnovni pojmovi



18/69

❑ Upravljanje protokom (Flow Control):

- upravljanje brzinom prijenosa podataka između predajnika i prijemnika.

❑ Detekcija greški (Error Detection):

- greške uzrokovane pojačavanjem signala, šumom.
- prijemnik otkriva prisustvo greški:
 - slanje signala predajniku za retransmisiju podataka.

❑ Ispravljanje greški (Error Correction):

- prijemnik identificira i korigira greške bitova (bit errors) bez potrebe za retransmisijom.



Otkrivanje i ispravljanje grešaka

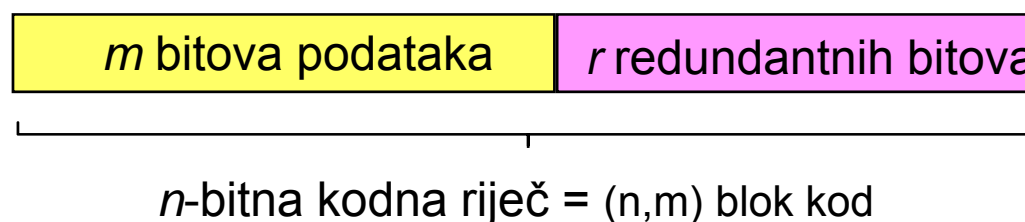
Otkrivanje (detekcija) greške

Tehnike otkrivanja grešaka zahtijevaju dodavanje redundantnih bitova podacima koji se prenose.

Redundancija signala: nadzor kvaliteta signala (samogašenje, prigušivanje,...).

Redundancija kodiranja: vremensko narušavanje rada dekodera.

Redundancija podataka: kodovi otkrivanja greški.



Kriterij kakvoće:

- Efikasnost koda: $CEF = m/n$,
- Hammingova udaljenost,
- Rezidualna brzina greške (Residual Error Rate).

Otkrivanje i ispravljanje grešaka

Rukovanje greškom

- U industriji se postavljaju visoki zahtjevi na integritet podataka.
- Kvaliteta prijenosa nije jako visoka, ali su posljedice gubitka podataka teške.
- Greške koje se ne otkriju (i sprave) u podatkovnom sloju teško je ispraviti u višim slojevima.

Dva načina borbe protiv grešaka:

1. Kodovi otkrivanja grešaka (Error Detecting Codes).

U svakom bloku podataka unosi se dovoljno redundancije da bi se mogla korigirati greška na prijemnoj strani (primjeri: bitovi pariteta (parity bits), suma provjere (checksum) i CRC (Cyclic Redundancy Check)).

2. Kodovi ispravljanja grešaka (Error Correcting Codes).

- Uljučuju dovoljno redundantnih bitova za otkrivanje i ispravljanje greške.
- Rijetko se koriste u industrijskim komunikacijskim mrežama (uglavnom u bežičnim komunikacijama i komunikacijama preko energetske kabele).
- Okviri sa greškama se odbacuju.



Otkrivanje i ispravljanje grešaka

Hammingova udaljenost

- Pomoću algoritama za računanje redundantnih bitova konstruira se lista valjanih kodnih riječi i iz te liste se nađu dvije kodne riječi koje imaju minimalnu HD. Tada je to HD cijelog koda i korekcija i detekcija greške se zasniva na njoj.
- Za detektiranje d (jednobitnih) greški (ZD) potrebno je imati kod sa HD od najmanje $d+1$ bita, zato što je onda nemoguće da greška u d bitova rezultira kodnom riječi koja je ispravna.
- Kad prijemna strana primi neispravnu kodnu riječ traži retransmisiju. Za korekciju d grešaka (ZC) potrebno je imati $2d+1$ Hammingovu udaljenost, što znači da su dvije ispravne kodne riječi toliko daleko da je čak i sa d greški originalna kodna riječ bliža neispravnoj nego druga kodna riječ.
- Pažljivim odabirom provjerenih bitova, moguće je dobiti kodnu riječ sa velikom Hammingovom udaljenosti.
- Što je veća HD, bolja su svojstva otkrivanja greški.





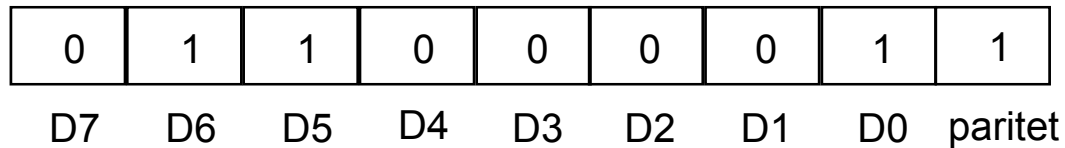
Otkrivanje i ispravljanje grešaka

Hammingova udaljenost – potreban broj bitova za ispravljanje greške

- Kako formirati kodnu riječ duljine $n=m+r$ bitova koja može ispravljati sve jednobitne greške?
- Sve podatkovne riječi su dopuštene, odnosno ima 2^m dopuštenih nizova.
- Svaka od 2^m legalnih poruka ima n ilegalnih kodnih riječi sa $HD=1$.
- Ove kodne riječi dobivaju se sistemskim invertiranjem svakog bita n -bitovne riječi.
- Dakle, svaka poruka zahtijeva $(n+1)$ nizova bitova za sebe (n -nizova koji se razlikuju za jedan bit i jedan niz koji je stvarna poruka).
- Ukupan broj nizova bitova je $(n+1) \cdot 2^m \leq 2^n$.
- Svih $(n+1) \cdot 2^m$ kodiranih poruka mora biti jednoznačno, pa ne može biti manje od 2^n kodnih riječi da bi se ispunio zahtjev za ispravljanje jednobitnih grešaka.
- Kako je $n=m+r$ može se pisati:
$$(m+r+1) \cdot 2^m \leq 2^{m+r}, \text{ odnosno } (m+r+1) \leq 2^r.$$
- Za zadani m ovaj izraz daje minimalan broj dodatnih bitova za otkrivanje i ispravljanje jednobitovnih grešaka.
- **Primjer:** za $m=32$ je $r=6$, $m=1000$ je $r=10$.

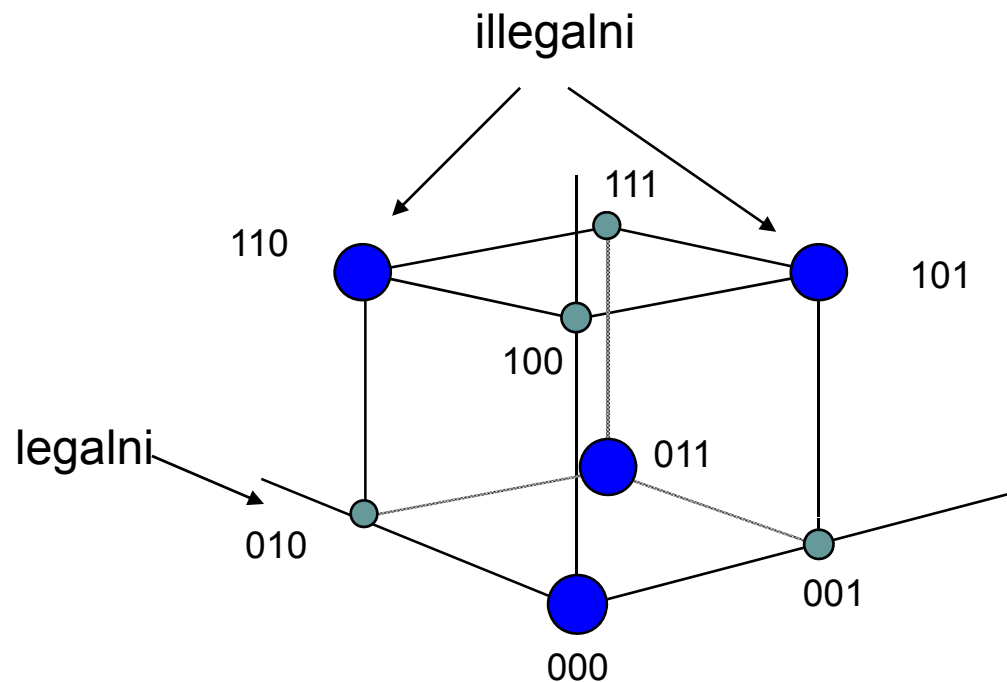
Otkrivanje i ispravljanje grešaka

Hammingova udaljenost u 3D prostoru - paritet



Neparni paritet: suma modulo-2 svih "1" u kodnoj riječi (uključujući bit pariteta) je neparna.

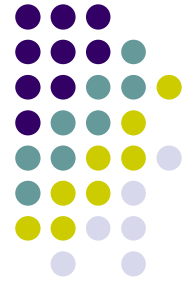
Bit pariteta je zadnji poslani bit (->LSB prvi, po konvenciji).



Otkrivanje i ispravljanje grešaka

Paritetni bitovi (Parity Bits)

- Primjer za kod s detekcijom greške bio bi dodavanje paritetnog bita kodnoj riječi.
- **Paritetni bit odabere se tako da ukupni broj bitova "1" bude paran (ili neparan).**
- Npr. na kodnu riječ 10110101 doda se 1 i to daje 10110101**1** ili na kodnu riječ 10110001 doda se 0 i dobije se kodna riječ 10110001**0**. Kod sa jednim paritetnim bitom ima udaljenost 2 (HD=2), što znači da se mogu detektirati pogreške u jednom bitu (čim se jedan bit promijeni, poremeti se parnost).
- Ako imamo kod sa samo 4 ispravne kodne riječi npr. 0000000000, 0000011111, 1111100000, 1111111111, udaljenost mu je 5 i mogu ispravljati 2-bitne pogreške.
- Kada bi npr. prijemnik dobio kodnu riječ 0000000111, znao bi da je original 0000011111. Sve greške na više od 2 bita ne mogu se popraviti.
- Kodovi za detekciju greške koriste se kad je isplativije ponovo poslati podatke nego ispravljati greške.



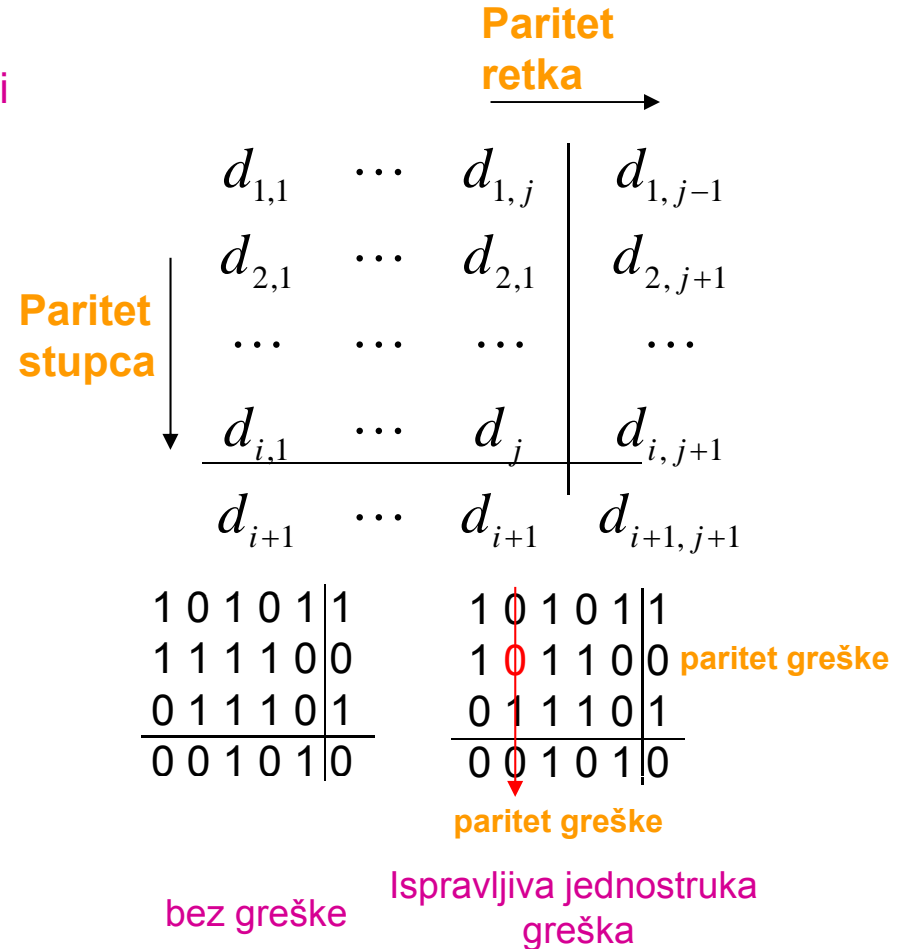
Otkrivanje i ispravljanje grešaka

Paritetni bitovi: primjeri

Jednodimenzionalni paritet bita pronalazi pojedinačne bitovne greške



Dvodimenzionalni paritet bita pronalazi i ispravlja pojedinačne bitovne greške



Otkrivanje i ispravljanje grešaka

Karakteristike kodova za detekciju grešaka

- Ispravljanje grešaka je relativno skupo (računski i gubitkom prijenosnog kapaciteta).
 - Potrebno je 10 redundantnih bitova za ispravljanje jednobitne greške u 1000-bitnoj poruci.
 - Ispravljanje u industrijskim komunikacijama samo iznimno.
 - Važnija je raspoloživost podataka na vrijeme, pa će uz njihovu povremenu neispravnost, nego ispravnost uz nedopustivo veliko kašnjenje prijenosa.
- Za otkrivanje jednobitnih grešaka dovoljan je samo jedan dodatni bit, bez obzira kako je duga poruka (paritet).
- Najpopularnija tehnika otkrivanja grešaka zasniva se na polinomskim kodovima ili CRC kodovima.

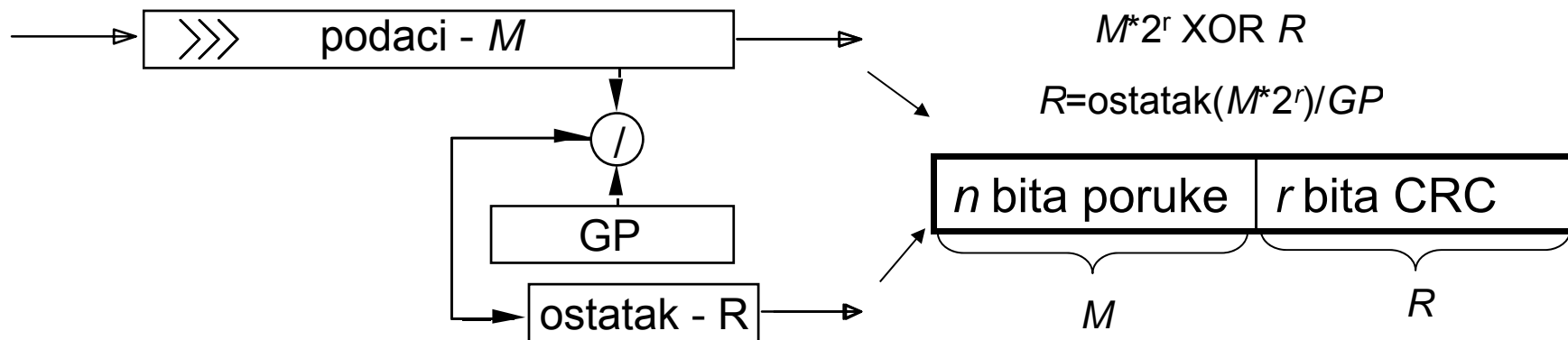


Otkrivanje i ispravljanje grešaka

CRC kod

Princip

- Niz podataka (okvir) sastoji se od n "0" i "1"
- Ovaj niz podataka se tretira kao polinom n -tog stupnja.
- Polinom se dijeli sa tzv. generator polinomom (GP) stupnja m , $m < n$,
- Ostatak dijeljenja (koji ima $(m-1)$ bitova) dodaje se na kraj niza podataka.



- Isto dijeljenje se obavlja i u prijemniku, ostatak dijeljenja mora biti identičan prenesenom ostatku.
- Generator polinom se odabira u skladu sa očekivanim smetnjama (greške na signalu i greške u modelu kanala (inverzija bitova))
- Široko rasprostranjen u praksi (ATM, HDCL).



Otkrivanje i ispravljanje grešaka

CRC kod

- Kodna riječ od n bitova predstavlja se kao polinom $n-1$ stupnja (od x^{k-1} do x^0), gdje su koeficijenti 0 i 1. Npr. kodna riječ 110101 ima 6 bitova i predstavljena je polinomom $x^5+x^4+x^2+x^0$.
- Kod CRC kodova predajnik i prijemnik se moraju dogovoriti o generator polinomu $G(x)$ (najviši i najniži bit generator polinoma mora biti 1).
- Okvir duljine m bita (predstavljen polinomom $M(x)$) mora biti veći od duljine generator polinoma.
- Na predajnoj strani se izračuna polje za provjeru pomoću generator polinoma i doda se na kraj okvira. Tako dobiveni okvir se na prijemnoj strani dijeli sa generator polinomom i ako nema ostatka znači da je okvir tačno prenesen tj. da nije došlo do greške prilikom prijenosa.

Postupak računanja polinoma koji će se prenijeti:

1. Pomnožiti $M(x)$ sa x^r tj. dodati r nula na kraj poruke što daje $T(x)$.
2. Podijeliti $T(x)$ sa $G(x)$ i zapamtiti ostatak
3. Oduzeti ostatak dijeljenja od $T(x)$



Otkrivanje i ispravljanje grešaka

CRC kod

- Ciklički kodovi detektiraju jednostruke i dvostruke greške i imaju visok stupanj otkrivanja greški u nizovima (sa r kontrolnih bitova detektirati će sve greške u nizovima čija je duljina $\leq r$).
- Ciklički kodovi su osjetljivi na greške tipa dodavanja ili oduzimanja bitova iz kodne riječi, te dodavanja niza nula.
- Primjeri generator polinoma su:

$$\text{CRC-12} = x^{12} + x^{11} + x^3 + x^2 + x^1 + 1$$

$$\text{CRC-16} = x^{16} + x^{15} + x^2 + 1$$

$$\text{CRC-CCITT} = x^{16} + x^{12} + x^5 + 1$$

$$\text{CRC-32} =$$

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$



Otkrivanje i ispravljanje grešaka

CRC kod

Primjer: računanje kontrolnih bitova

$M(x) = x^7 + x^4 + x^3 + x^1$ ili 10011010 je originalna poruka.

$r = 3$ stupanj generator polinoma $G(x) = x^3 + x^2 + 1$ tj. 1101

Množenjem $M(x)$ sa x^3 dobiva se 10011010000 tj. $T(x)$.

Dijeljenje $T(x)$ sa $G(x)$:

$$10011010000 : 1101 = 11111001$$

1101

1001

1101

1000

1101

1011

1101

1100

1101

1000

1101

101 - ostatak



Otkrivanje i ispravljanje grešaka

CRC kod

Oduzimanje ostatka od T(x) (XOR operacija):

$$\begin{array}{r} 10011010000 \\ - \quad \quad \quad 101 \\ \hline 10011010101 \end{array}$$

(dobije se isto kao da smo na originalnu poruku dodali ostatak dijeljenja)

Prenosi se bitovna sekvenca 10011010101, koja se na prijemnoj strani dijeli sa G(x).

Ako je ostatak tog dijeljenja 0 bitovi su dobro preneseni, u protivnom je došlo do greške kod prijenosa.

Provjera (dijeljenje prenesene bitovne sekvence sa generator polinomom na prijemnoj strani):

$$\begin{array}{r} 10011010101 : 1101 = \\ \underline{1101} \\ 1001 \\ \underline{1101} \\ 1000 \\ \underline{1101} \\ 1011 \\ \underline{1101} \\ 1100 \\ \underline{1101} \\ 1101 \\ \underline{1101} \\ 0000 \text{ - ostatak} \end{array}$$



Otkrivanje i ispravljanje grešaka

R_{er} (Residual Error Rate)

R_{er} = vjerojatnoća pojavljivanja nedetektirane greške u kodu za detektiranje greške kao funkcije vjerojatnoće pojave greške bita.



Primjer:

HD

2

Vjerojatnoća greške bita
(Bit error probability)

E_r

R_{er}

$$R_{er} = 1 - (1 - E_r)^n - n \cdot E_r \cdot (1 - E_r)^{n-1}$$

nema greške

tačno jedna greška

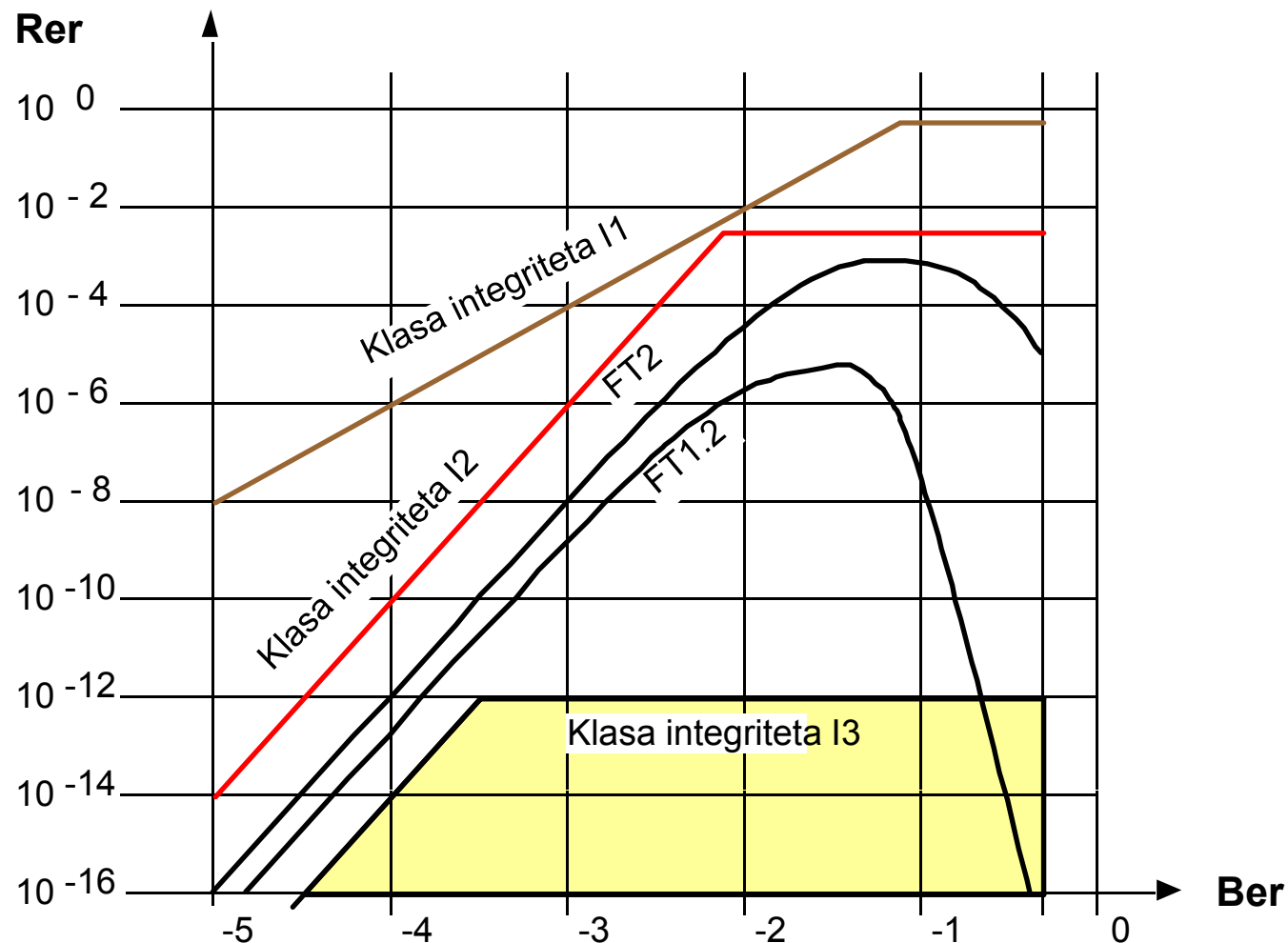
R_{er} za duljinu dvije riječi:

$E_r = 10^{-5}$	n = 9 bit	$R_{er} = 72 \cdot 10^{-10}$	efikasno....
$E_r = 10^{-5}$	n = 513 bit	$R_{er} = 2.6 \cdot 10^{-5}$	nekorisno ...

Otkrivanje i ispravljanje grešaka

Klase integriteta i Ber (bit error rate)

Standard IEC 61870-5 (protokoli za daljinsko upravljanje podstanicama) definira nekoliko klasa robusnosti kao funkcije Ber-a (loši/dobri bitovi).



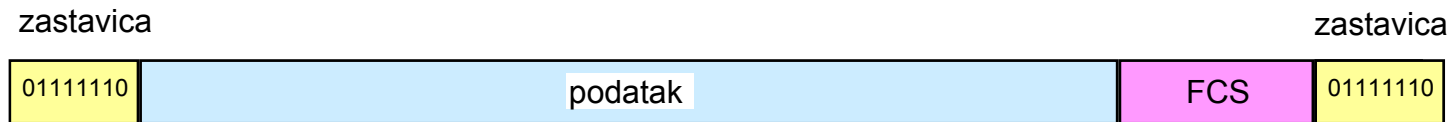
Otkrivanje i ispravljanje grešaka

Sinhronizacija greški

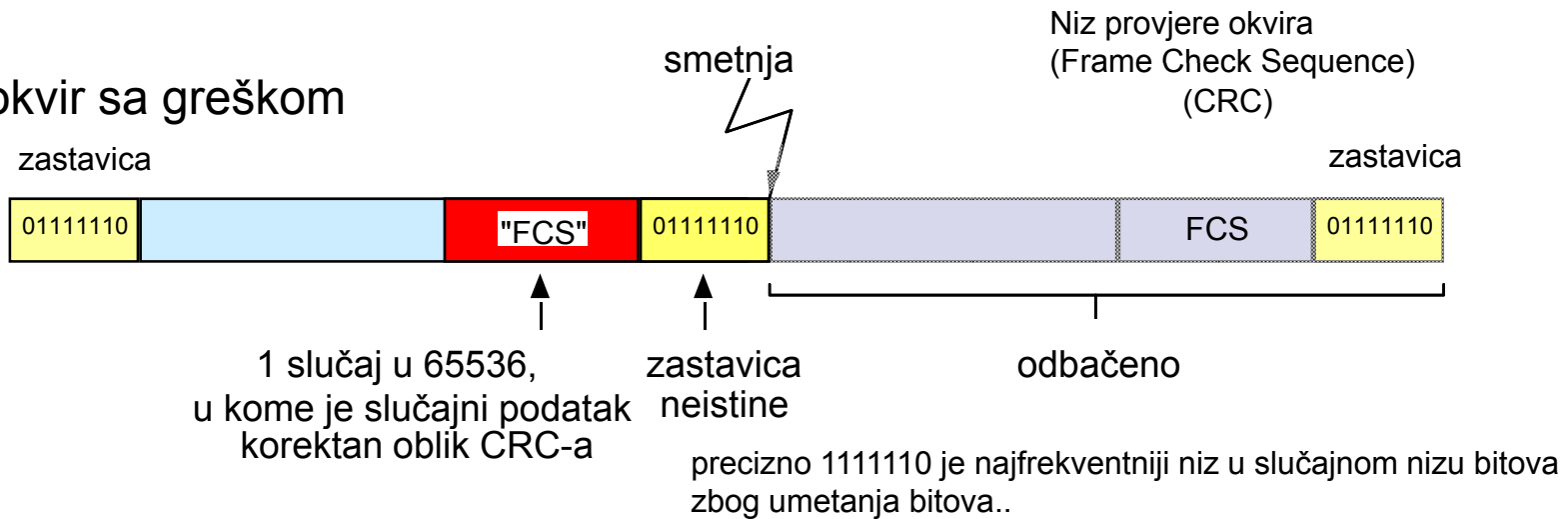


35/69

HDLC-okvir



HDLC-okvir sa greškom



Pojedinačna greška može pokvariti okvir -> HD = 1

Nije interesantno kako izgleda ovakav slučaj, već kako se on može dogoditi.

Zbog ovog buga, HDLC koji se koristi u industriji zahtijeva dodatne provjere greške.

Sinhronizacija treba sadržavati veći HD nego što je duljina podatka.

Otkrivanje i ispravljanje grešaka

Kodovi za ispravljanje grešaka

- Kodovi za ispravljanje (korekciju) grešaka koriste se kada postoji **velika vjerojatnoća pojavljivanja smetnje** (npr. komunikacija energetskim kabelom) i kada je **vrijeme retransmisije dugo** (npr. ispitivanje prostora blizu planete Jupiter).
- U industriji su kodovi za korekciju greške direktno ugrađeni u fizičkom sloju, npr. kao dio višetonskog prijenosa (ADSL) ili radio protokola (Bluetooth).
- Ispravljanje grešaka neminovno smanjuje (narušava) integritet podataka, tj. vjerojatnoću prihvaćenih netačnih podataka, budući da je redundancija korekcije uzeta iz redundancije koda.
- Mnogo je važnije uklanjanje pogrešnih (netačnih) podataka (nizak Rer) nego pokušaj korekcije prijenosa.
- Međutim, kada je prijenosni medij veoma loš (radio), korekcija greški je neophodna nakon svake kratke prenesene poruke.
- Pridruženi bitovi za detekciju i korekciju greške je kompromis između integriteta i raspoloživosti.



Otkrivanje i ispravljanje grešaka

ECC (Error Control Coding)



- Detektiraju i nastoje ispraviti greške koje nastaju pri prijenosu u digitalnim komunikacijskim sistemima.
- Enkoder osim informacijskih šalje i redundantne bitove.
- Dekoder koristi redundantne bitove za detekciju i moguću korekciju grešaka.
- Dvije vrste kodiranja:
 - Blokovo (BCH, Reed-Salomon, cikličko, itd.)
 - Konvolucijsko (konvolucijsko kodiranje i Viterbi dekodiranje).
- Blokovo kodiranje – preslikavaju fiksni broj simbola poruke u fiksni broj simbola koda.
- Koder bloka podataka razmatra svaki blok kao neovisan i predstavlja memorijsku jedinicu.
- Svaki blok podataka sastavljen od m simbola kodira se u kodnu riječ duljine n simbola.
- m = duljina poruke, n = duljina kodne riječi, oznaka koda $[n, m]$.

Otkrivanje i ispravljanje grešaka

BCH kodiranje

- Poruka za $[n,m]$ BCH kod mora biti m -ti stupac binarnog Galois polja.
- Kod koji odgovara ovoj poruci je n -ti stupac Galois polja.
- Svaki redak Galois polja predstavlja jednu kodnu riječ.
- Parametri BCH koda:
 - n , duljina kodne riječi, cjelobrojnik oblika 2^k-1 za neke cjelobrojne vrijednosti $k > 2$.
 - m , duljina poruke, pozitivan cijeli broj manji od n . Međutim samo neki pozitivni cijeli brojevi manji od n su validni za izbor m .
 - parametar t , sposobnost korekcije greške.

Primjeri: $n=7$ i $n=15$

n	m	t
7	4	1

n	m	t
15	11	1
15	7	2
15	5	3

n može biti 7,15, 31, 63, 127, 255, 511



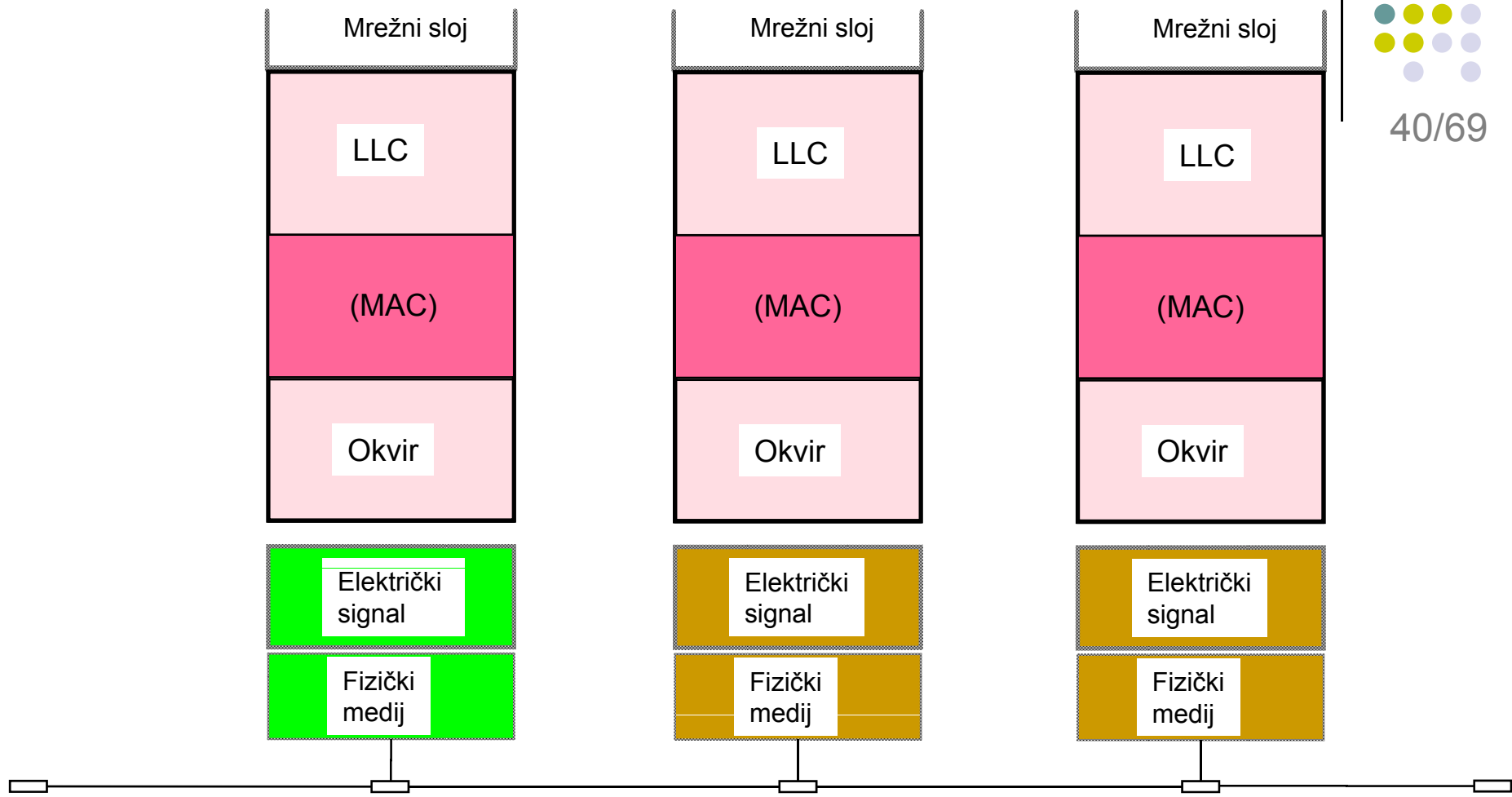
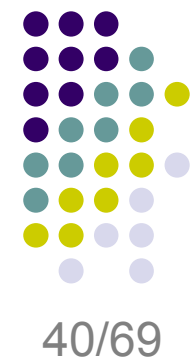
Otkrivanje i ispravljanje grešaka

BCH kodiranje

- Galois polje je algebarsko polje koje ima konačan broj elemenata.
- Ovo polje ima 2^k članova i koristi se u kodiranju sa upravljanjem greške (error-control coding) i označava se sa $GF(2^k)$.
- Samo Reed-Solomon kodiranje koristi $GF(2^k)$, dok ostali, uključujući i BCH, koristi $GF(2)$.
- **Primjer: računanje elemenata polja $GF(2^3)$.**

Cjelobrojni prikaz	Binarni prikaz	Element $GF(8)$
0	000	0
1	001	1
2	010	A
3	011	A+1
4	100	A^2
5	101	A^2+1
6	110	A^2+A
7	111	A^2+A+1

5.3. Podsloj pristupa mediju



Podsloj pristupa mediju (Medium Access Control) daje pravo stanici slanja podataka u višemasterskoj sabirnici.

Kriterij kvalitete



41/69

Jednakopravnost
(Fairness)

Sve stanice će dobiti pravo pristupa mediju.

Determinizam
(Determinism)

Svim će stanicama biti omogućen pristup mediju unutar konačnog vremena.

Pravovremenost
(Timeliness)

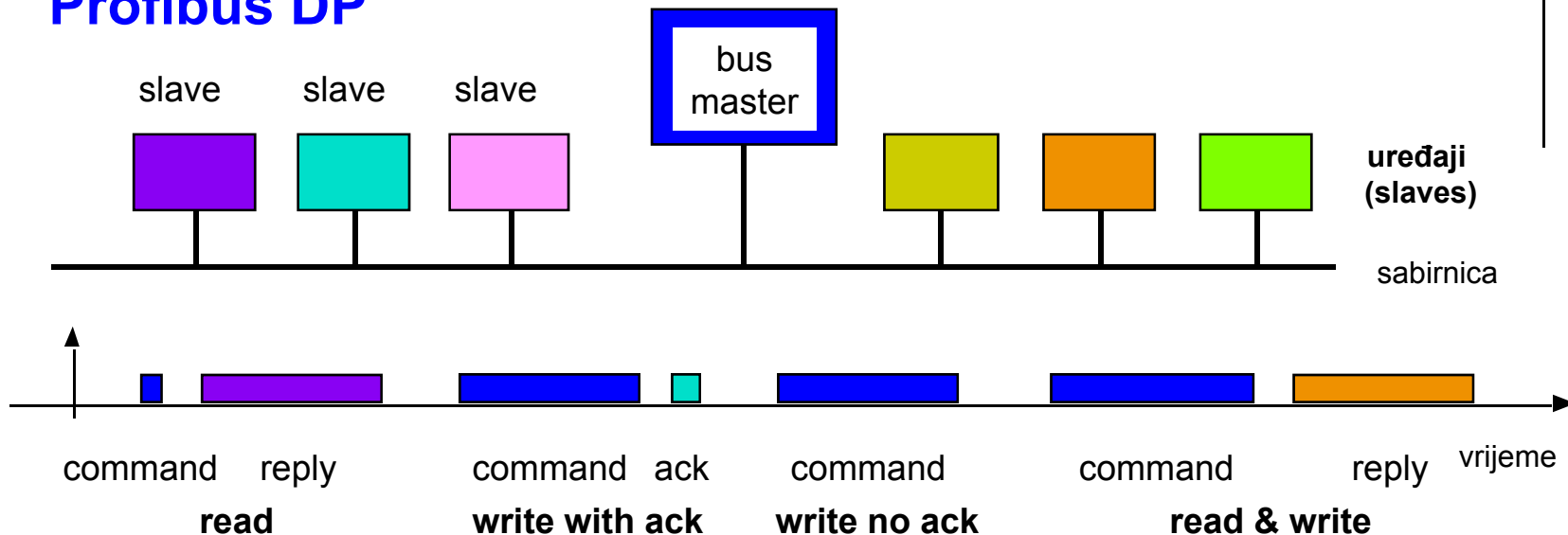
Svim će stanicama biti omogućeno poslati okvir unutar određenog vremenskog intervala, ovisno o njihovu prioritetu.

Robusnost
(Robustness)

Komunikacijske greške ili trajni kvar neke stanice neće ometati ostale stanice u pristupu mediju.

MAC pojedinačni master

Profibus DP



Zadaci mastera:

- Dodjeljuje vremenski položaj, odnosno mjesto svakog slave-u na sabirnici.
- Može pridodjeljivati prioritete (ili neprioritete: round-robin, svi se jednako tretiraju).
- Može biti izvor i odredište podataka.

Prednost: striktno deterministički, kompletno i fleksibilno upravljanje.

Nedostaci:

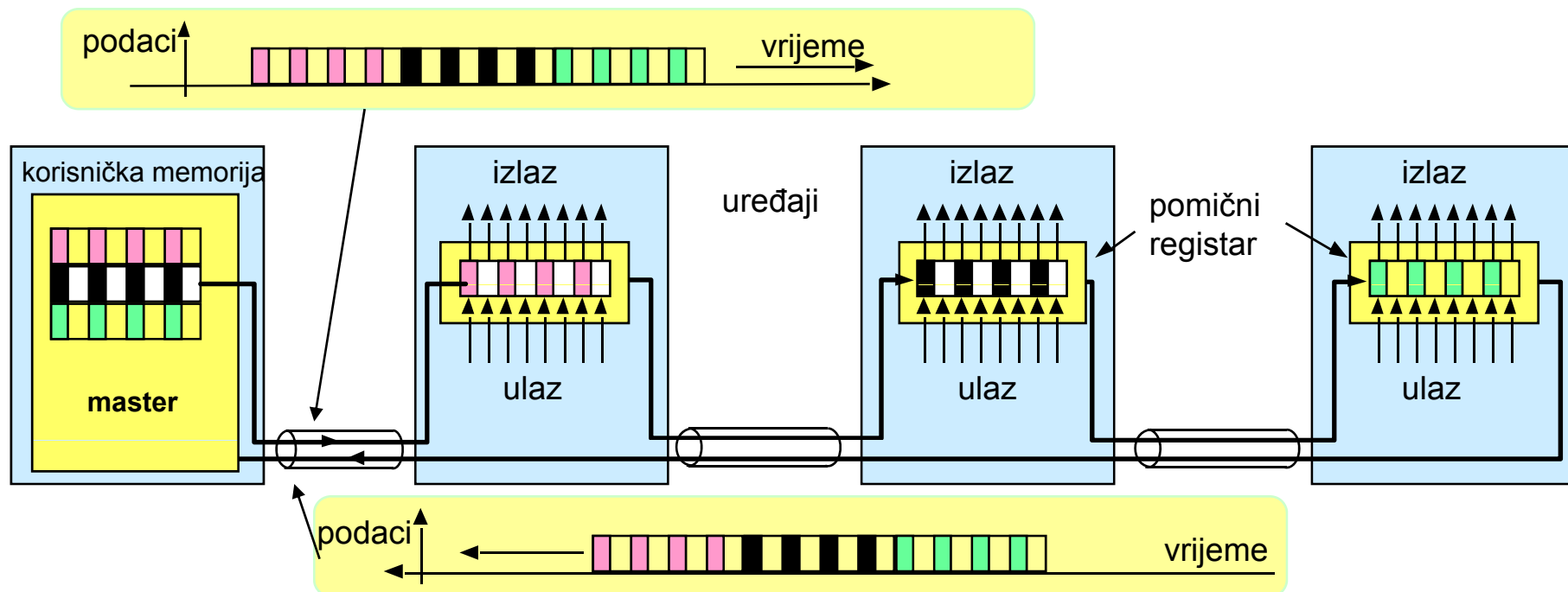
- Polling uzima vrijeme, budući da uređaji koji nemaju prijenosa moraju mu biti podvrgnuti.
- Master posjeduje oskudno znanje o važnosti podataka.

Poboljšanje: "look-at-me" (kratak okvir pollinga dopušta slave-u da zahtijeva polling većeg okvira) = "inicijativa slave-a" korištena u Profibus DP.

MAC prstenovi

Princip umetanja registra

- Uređaji su povezani tačka-tačka linijama (ne sabirnicama!), postoji jedan predajnik po segmentu.
- Operacija je slična onoj koju ima veliki pomični registar.
- Master šalje duge okvire zajedno sa izlaznim podacima do prvog uređaja u prstenu.
- Svaki uređaj čita podatke koji su mu namijenjeni, umjesto da ih upisuje on prosljeđuje (propušta) okvir do sljedećeg uređaja.
- Zadnji uređaj šalje okvir nazad prema masteru.



MAC prstenovi

Princip umetanja registra

Dva najvažnija fieldbusa korištena u prstenastoj topologiji su **Interbus-S** i **SERCOS** koji koriste opisani princip umetanja registra.

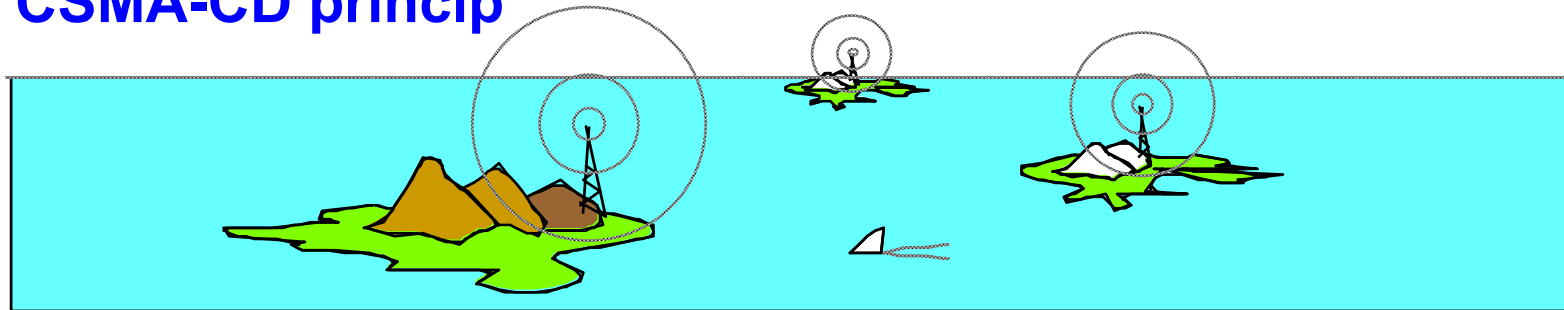


44/69

- **Pozicija bita u okviru odgovara poziciji uređaja u prstenu.**
 - **Ne postoje adresirani uređaji – jednostavno za korištenje.**
 - **Svaki uređaj unosi kašnjenje od nekoliko bita.**
- + deterministički pristup, dobro korištenje kapaciteta, adresiranja su dana sa nizom uređaja u prstenu, samo sa tačka-tačka linijama.
- veliko kašnjenje (nekoliko μ s po uređaju).

MAC Ethernet

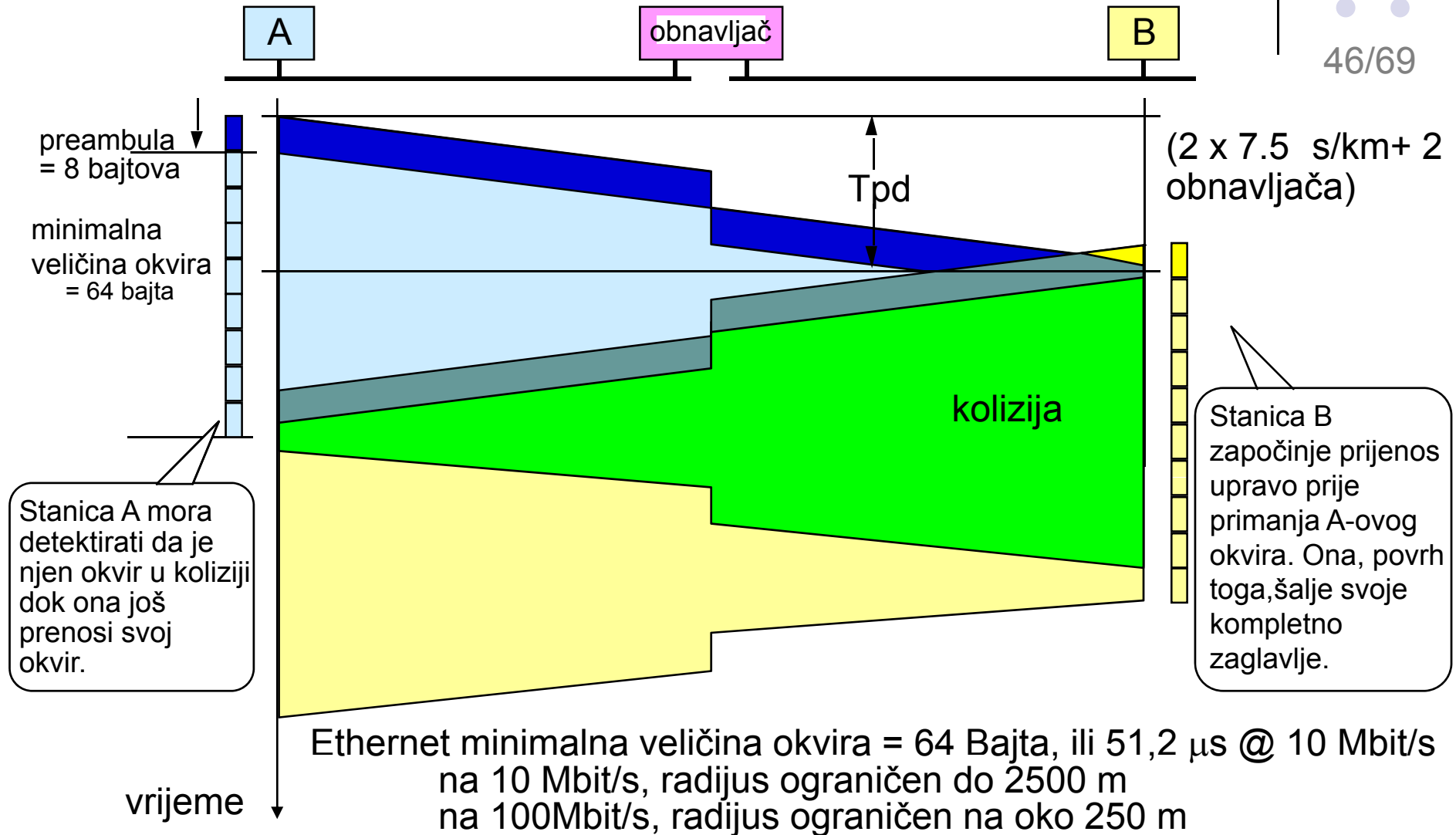
CSMA-CD princip



- Princip Svaka stanica šalje po nađoenju, ako nema potvrde, izvršiti retransmisiju. (čista Aloha)
- Poboljšanje 1:** Ne slati kada je medij zauzet. (osjetljiv nosilac)
- Poboljšanje 2:** Biti svjestan gužve. (detekcija kolizije)
- Poboljšanje 3:** Ponoviti nakon slučajnog iznosa vremena, duplirati nakon svake kolizije, maksimalno 15 puta. (binarni izlaz)
- Prednost: Arbitracija ne ovisi o broju ili adresama stanica.
- Nedostaci: Nema gornje granice za vrijeme čekanja, što znači vrijeme čekanja ovisi o brzini pristizanja okvira i njihovoj prosječnoj duljini.
- Pristup mediju nije deterministički, ali za semafore (<1%) ne postoji uočljivo kašnjenje.

MAC Ethernet

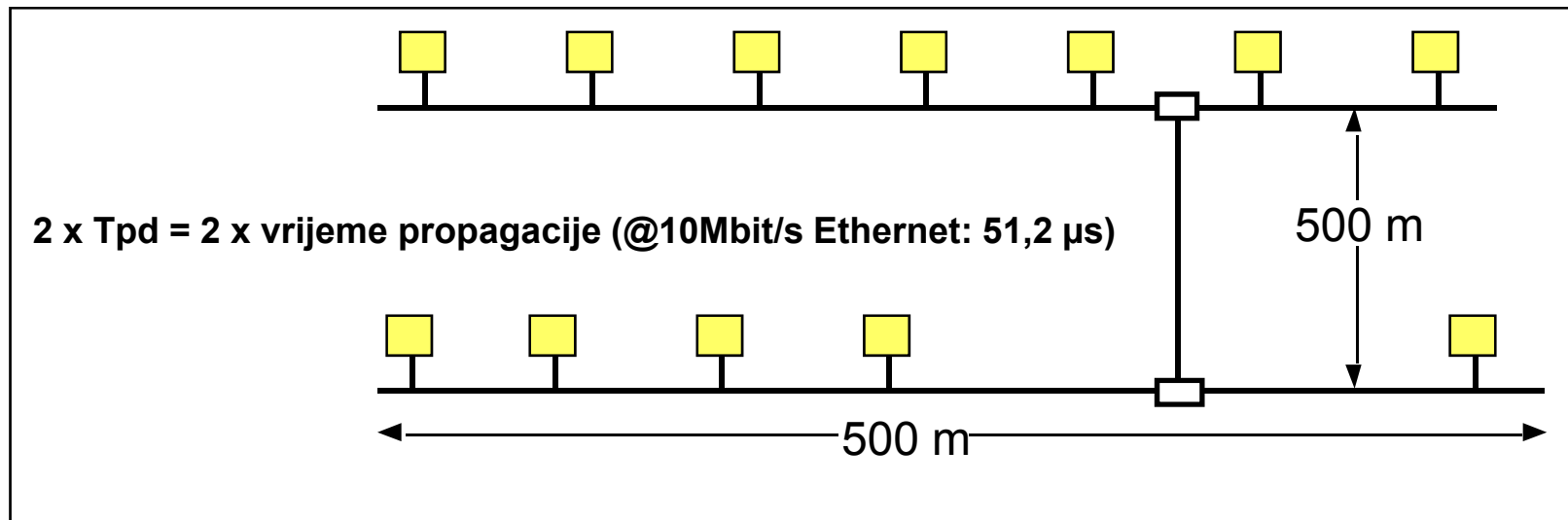
Uvjeti kolizije



MAC Ethernet

Uvjeti propagacije i dijametar sabirnice

Okvir mora biti dovoljno velik (dug) što omogućuje svim stanicama detekciju kolizije, dok je okvir u procesu prijenosa.



Kolizija se može detektirati kada duljina okvira nije duža od propagacijskog kašnjenja -> smanjenje na minimalnu veličinu (512 bita = 64 bajta).

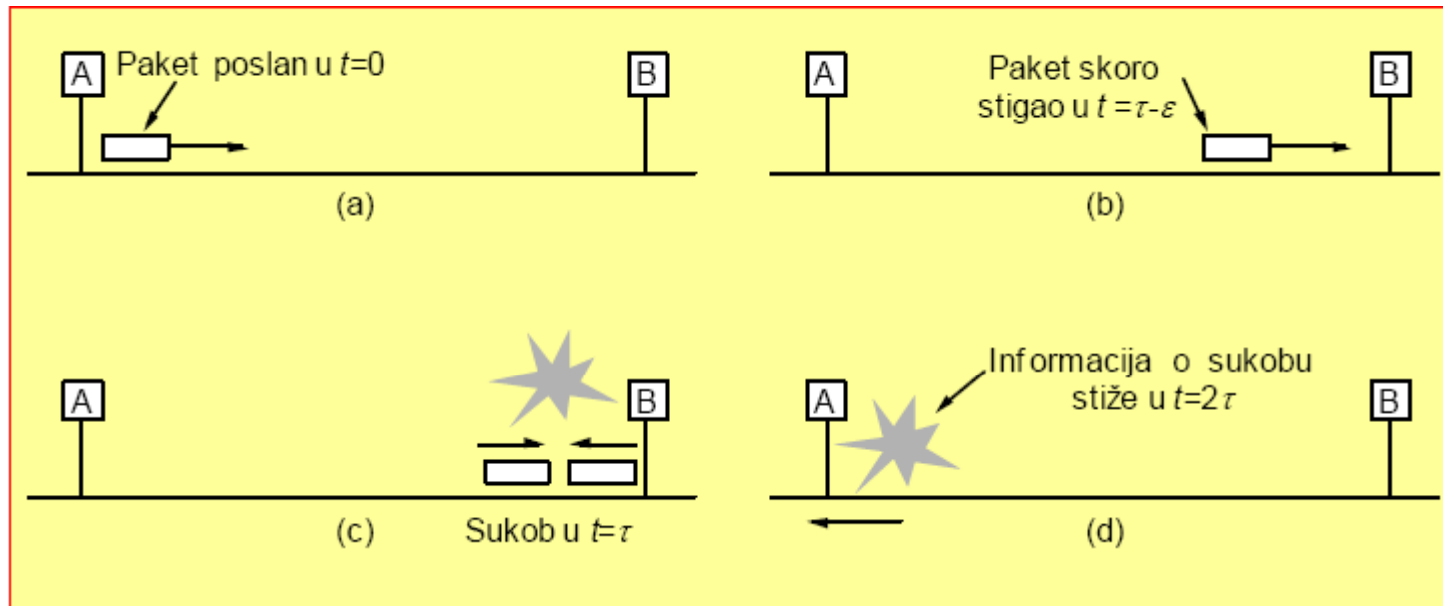
“Dijametar” mreže je ograničen na 2,5 km.

Budući da stanica koja očekuje koliziju mora čekati jedno vremensko mjesto prije prijenosa, maksimalna propusnost okvira na Ethernetu je ograničena vremenskim položajem.

MAC Ethernet

Uvjeti propagacije i dijametar sabirnice

Zašto je potrebna minimalna duljina okvira?



- Pad osigurava da prijenos traje dovoljno dugo da se poruka još uvijek šalje kada prvi bit stigne do prijemnika.
- Slanje okvira poruke mora još uvijek trajati kada do izvora stigne smetnja uzrokovana sukobom sa porukom druge stanice na mreži. =>
- **Minimalna duljina okvira 2τ .**

MAC Ethernet

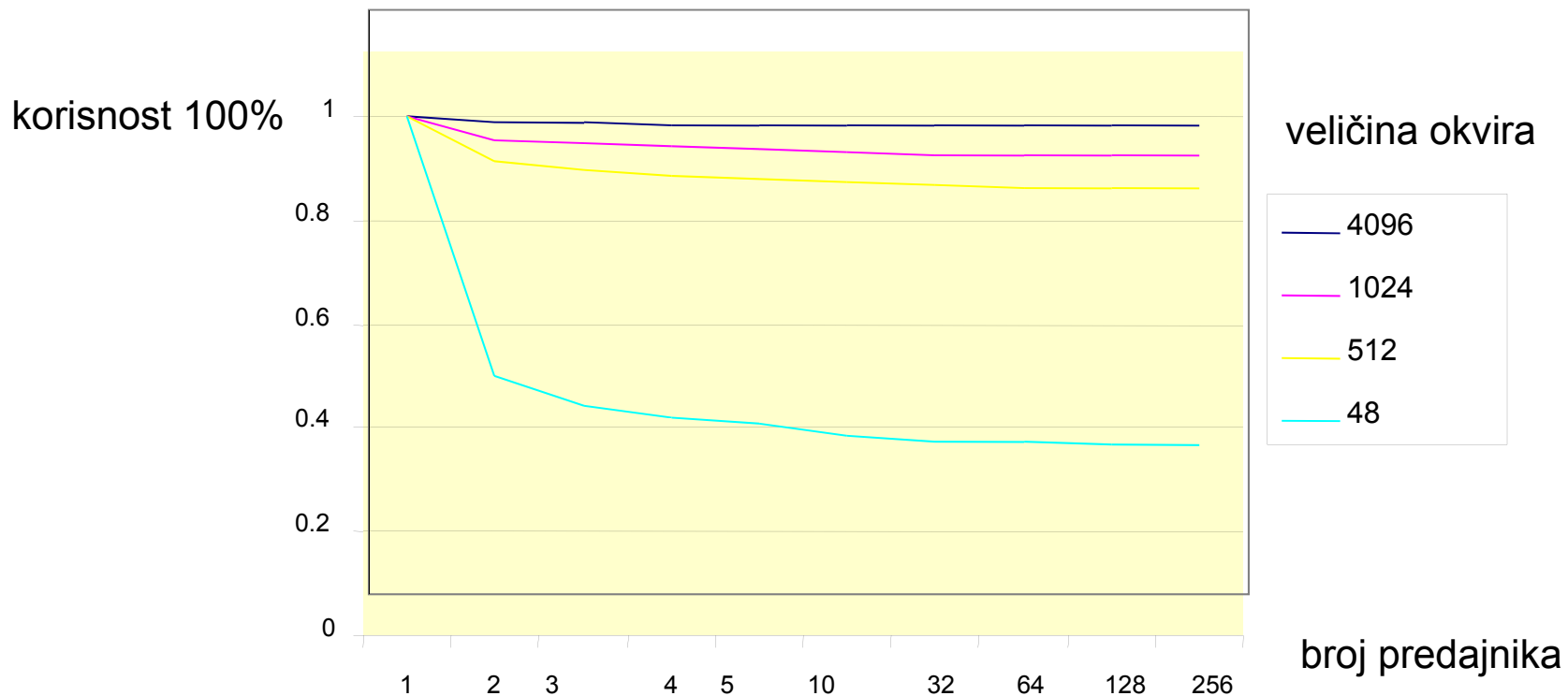
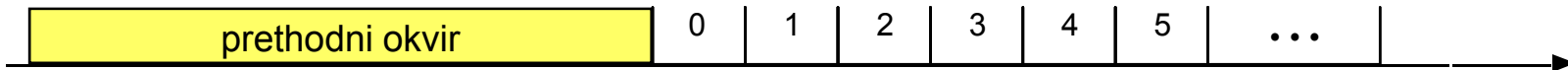
Vjerojatnost kolizija i istovremenost predajnika

Nakon kraja okvira, predajnik odabire slučajno slot iz fiksnog skupa slotova.



49/69

vrijeme



Ethernet nije efikasan za okvire male veličine i veliki broj istovremenih predajnika.

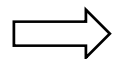
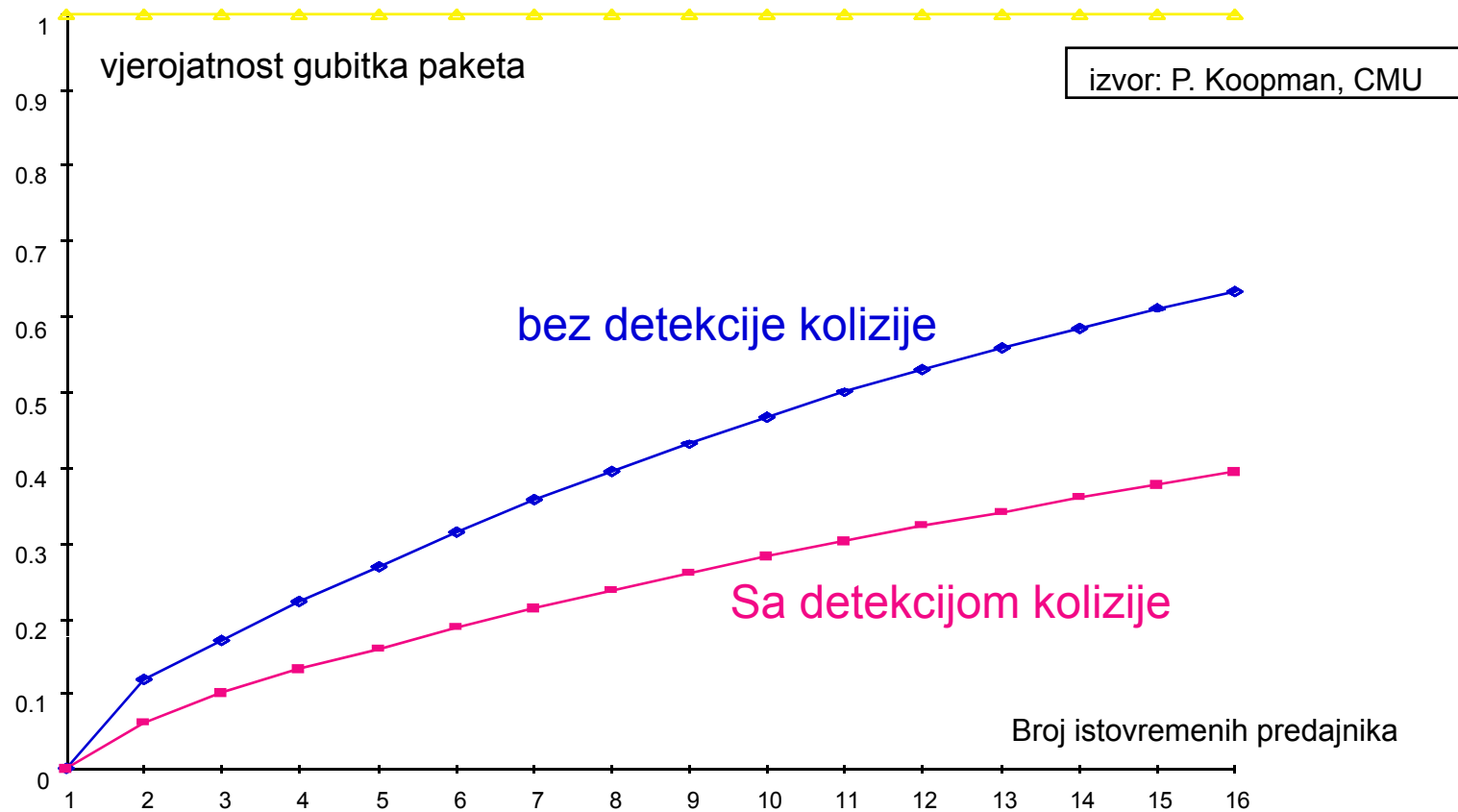
Ethernet se razmatra s ciljem unošenja preopterećenja kada se dosegne 10%-18% opterećenja (tereta).

MAC Ethernet

Kada se kolizija ne može detektirati?

Nije uvijek moguće detektirati koliziju.

LON koristi p-perzistentni MAC sa 16-slotnim izbjegavanjem kolizije ($p = 1/16$).



Veliki broj istovremenih predajnika uzrokuje visoku vjerojatnost gubitka paketa. LON može izvršiti ponovni pokušaj 255 puta: vjerojatnost gubitka poruke je mala, ali je kašnjenje veliko.

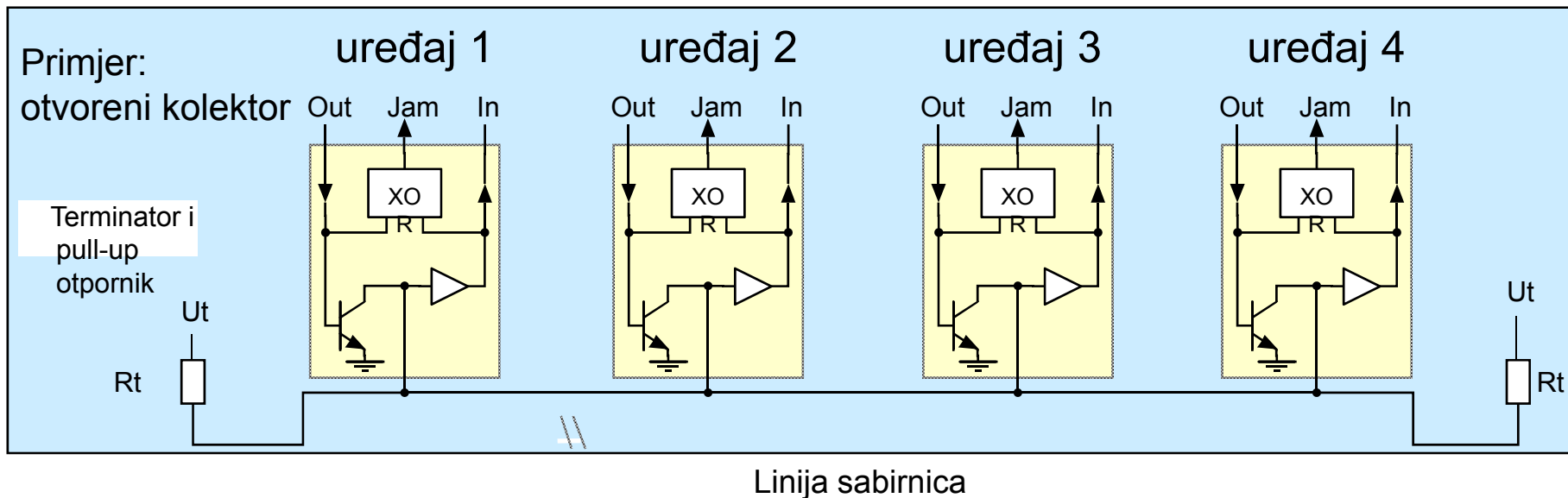
MAC CAN

Deterministička arbitraža

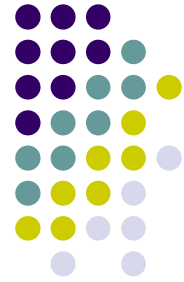
CAN fieldbus koristi medij sa dominantnim i recesivnim stanjem.

Kada je nekoliko predajnika istovremeno aktivno, dominantno stanje nadvladava recesivno stanje ako postoji najmanje jedan predajnik koji šalje dominantno stanje (dominantno je "Nisko" u otvorenom kolektoru, "Svjetao" u optičkom vlaknu, ili kolizija na mediju koji dopušta koliziju).

Takav medij dozvoljava bit-wise "Wired-OR" operaciju.



Uređaj je sposoban znati da li je signal koga je on poslao na liniju poremećen (XOR).

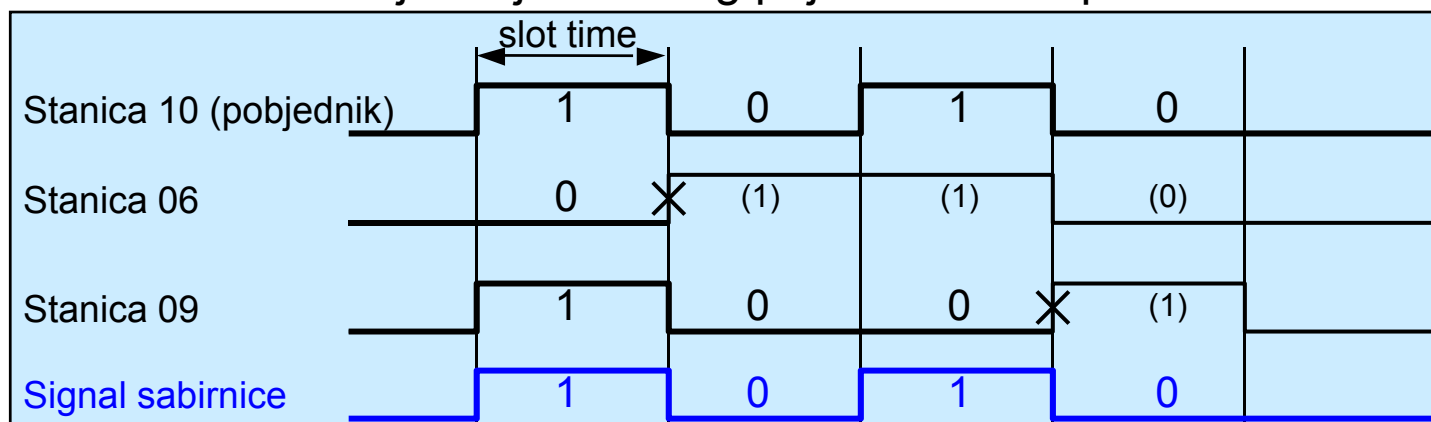




MAC CAN

Kolizija sa pobjednikom (Urn ili binarna bisekcija)

- Svaka stanica ima različit identitet (u ovom primjeru 4 bitni ID)
- Svaka stanica sluša prije slanja i može detektirati koliziju.
- Nadmetanje stanica započinje istovremenim slanjem (prvi bit je SYNC-predznak)
- Svaka stanica počinje svoj prijenos sa svojim identitetom, bit po bit.
- U slučaju kolizije, "1" pobjeđuje "0" ("1" = nizak, svjetao, dominantan).
- Stanica, čiji "0" bit je transformiran u "1" odmah se povlači.
- Stanica pobjednik nije poremećena i obavlja prijenos.
- Poražene stanice čekaju kraj trenutnog prijenosa da bi ponovo startale.



Prednost: deterministička arbitraža (pretpostavlja nepristrasnost), dobro ponašanje u prisustvu opterećenja.

Nedostatak: veličina jedinstvenog ID definira vrijeme arbitraže, kašnjenje prijenosa definira slot time -> 40m @ 1 Mbit/s, 400m @ 100 kbit/s.

MAC CAN

Deterministička arbitraža

Prednost: deterministička arbitraža (pretpostavlja nepristrasnost, to jest uređaj započinje ponovo slanje samo ako se desi da su svi uređaji gubitnici),
dobro ponašanje u prisustvu opterećenja.

Nedostatak: slot time (vrijeme jednog bita) mora biti dovoljno dugo da bi svaka stanica mogla detektirati da je došlo do poremećaja, to jest dva puta dulje od vremena propagacije sa jednog kraja sabirnice do drugog (brzina signala = 5 ns / m).

Dakle, brzina prijenosa bita ovisi o produženju sabirnice:

40m @ 1 Mbit/s, 400m @ 100 kbit/s

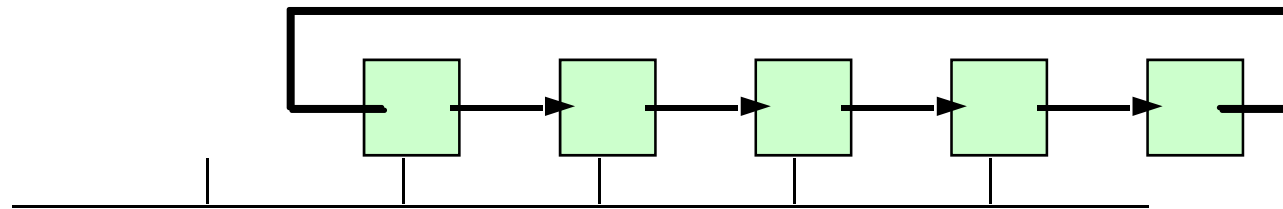
veličina jedinstvenog ID definira vrijeme arbitraže.



MAC Profibus

Princip zaloga

Token Bus (IEEE 803.4), Profibus (IEC 61158)



Sve stanice formiraju logički prsten.

Svaka stanica poznaje sljedeću u prstenu (i sljedeću iza nje)

Svaka stanica delegira pravo za slanjem sljedećoj stanici u prstenu.
(u obliku okvira zaloga ili kao dodatak na okvir podataka).

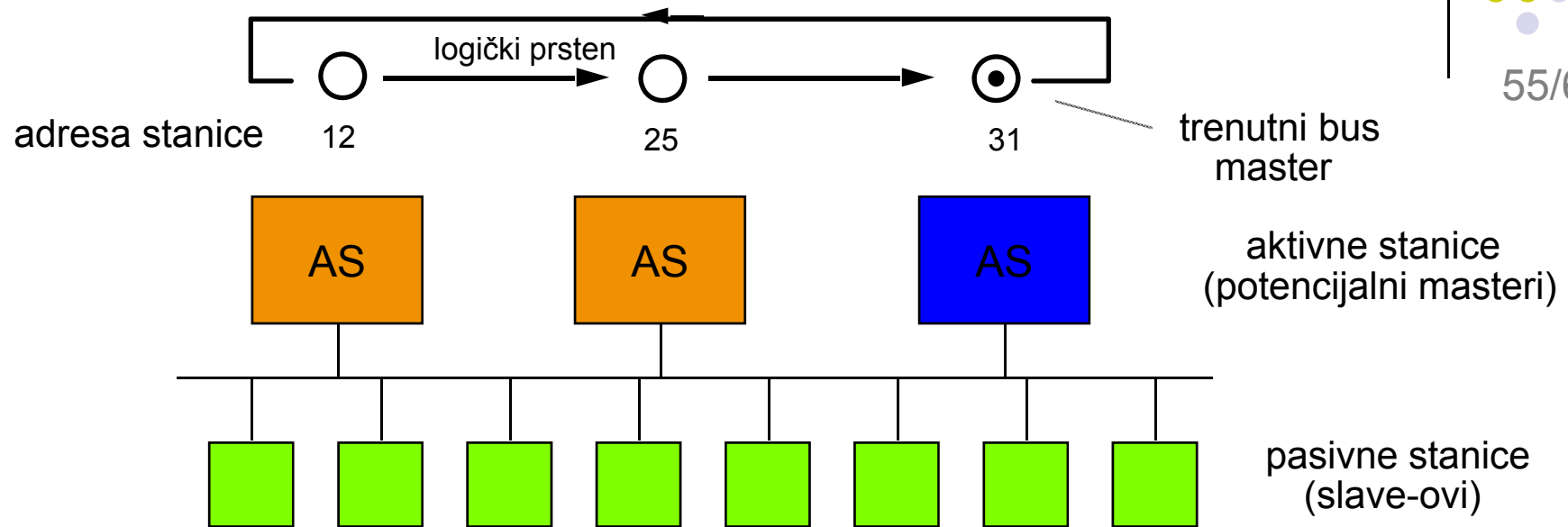
Varijante Zalog sa prioritetom (povratak na stanicu sa najvećim prioritetom)

Problemi: Gubitak ili dupliciranje zaloga, inicijalizacija.

Ne miješati sa token ringom!

MAC Profibus

Predaja zaloga



Aktivne stanice (AS) mogu postati masteri ako posjeduju vlastiti zalog, ograničenog trajanja (samo jedan okvir).

Nakon tog vremena, master predaje zalog stanici sa većom adresom, ili ako ima najveću adresu, stanici sa najmanjom adresom.

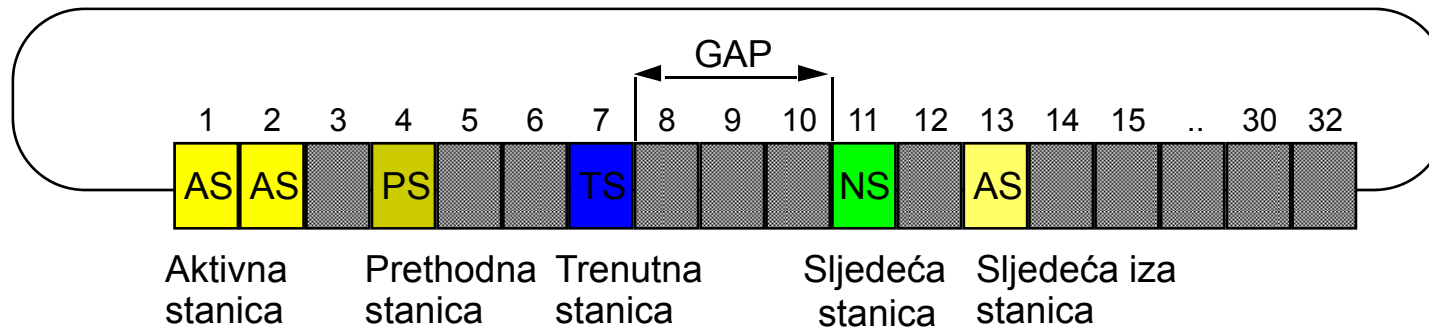
Stanica mora poslati najmanje jedan okvir (podatak ili zalog) kada dođe na red.

MAC Profibus

Algoritam predaje zaloga



Svaka stanica pamti listu aktivnih stanica (LAS)



Kada trenutni master nema više podatka za slanje, ili kada je prošao sve korake, šalje okvir zalog sljedećoj stanici (NS) u prstenu.

NS potvrđuje prijem zaloga.

Ako master ne primi potvrdu za dva uzastopna trial-a, master uklanja stanicu iz LAS-a i prijavljuje "sljedeću iza aktivnu stanicu" (OS) kao (NS).

Stanica prihvata zalog ako ga je primila dva puta.

Master testira regularne intervale sa "Request FDL-Status" za prisustvo budućih stanica u praznini između sebe i NS.

MAC Profibus

Inicijalizacija zaloga



Početna stanica “preslušava” sabirnicu prije pokušaja slanja podataka.

Ako osjeti promet, stanica “snima” okvire zaloga i gradi listu aktivnih stanica (LAS).

Posebno, ona promatra da li već postoji stanica sa istim imenom kao što je njeno.

Kada master provjerava prazninu (gap), stanica će primiti ponuđeni mu zalog.

Ako stanica ne registrira nikakav promet tokom izvjesnog vremena, ona sama sebi šalje okvir zaloga.

Ona šalje okvir prvog zaloga sebi, dozvoljavajući drugim startnim stanicama da se registriraju.

Samo kada ne detektira druge stanice tada ona započinje sa sistematičkim ispitivanjem svih adresa, da bi se izgradila LAS.

Usporedba MAC metoda



58/69

stohastičke

optimizam

prisustvo nosioca

detekcija kolizije

p-perzistentna kolizija

determinističke

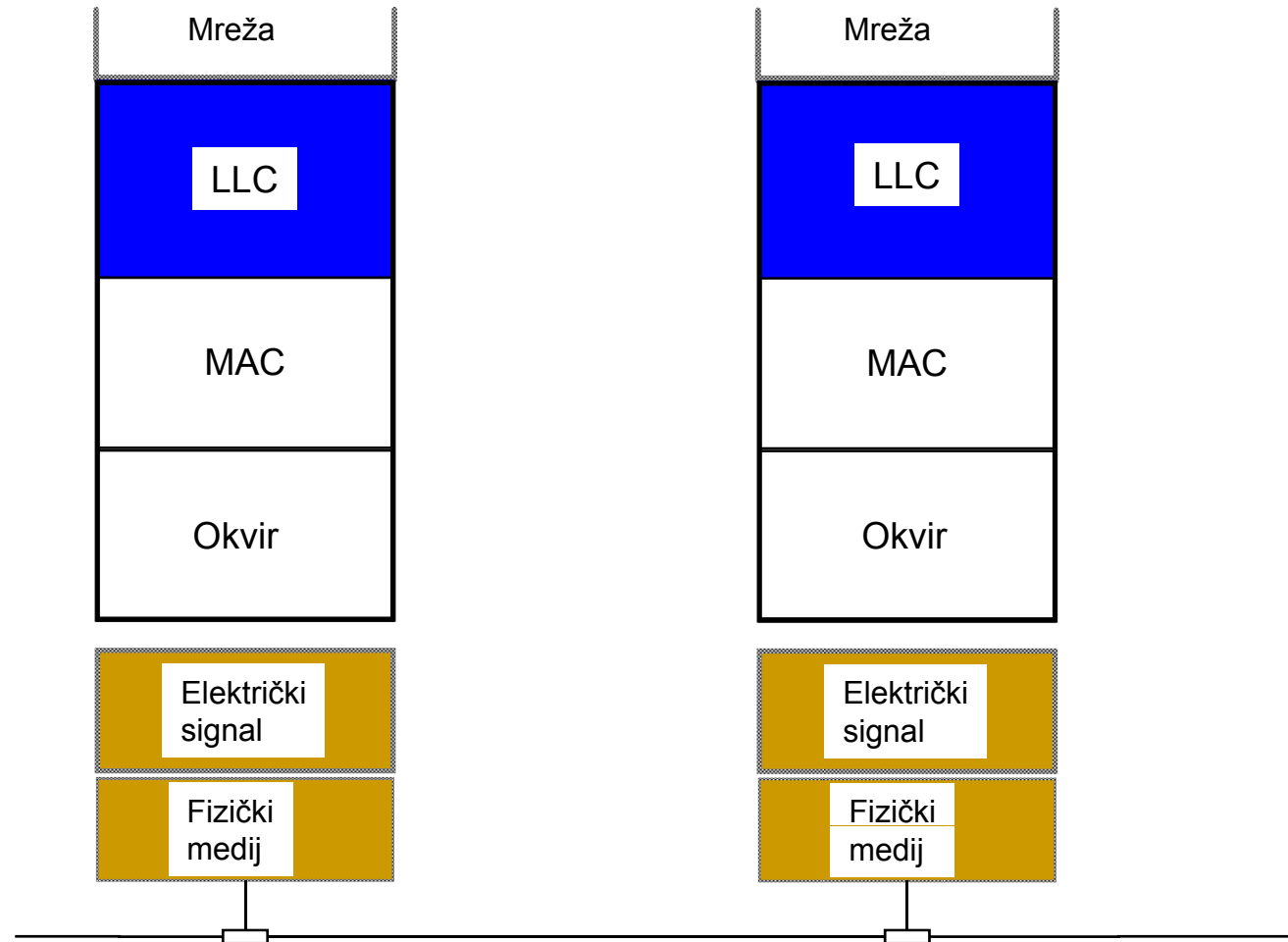
pesimizam

centralni master

kolizija sa pobjednikom

predavanje zaloga

5.4. Podsloj logičkog sloja



Dvije vrste usluga:

- bespojne usluge,
- spojno-orijentirane usluge.

Podslaj logičkog sloja

Ova razmatranja su primjenjiva na svim razinama OSI modela.

Bespojni način

(Datagram \approx pismo)

Svaki paket (Datagram) sadrži sve informacije potrebne za njegovo prosljeđivanje do ciljane lokacije u mreži, uključujući stazu za povratak natrag ka predajniku.

Mreža ne preuzima odgovornost za redoslijed paketa i ne pokušava ispraviti oštećene datagrame.

Teret upravljanja protokom i rekonstrukcija greške je premješten na aplikacijski sloj.

Semantika CL-prijenosa

```
Send_Packet (source, destination, Packet1);
```

```
Send_Packet (source, destination, Packet2);
```

Spojno-orijentirani način

(Virtualni krug \approx telefon)

Prvo se uspostavlja konekcija između predajnika i prijemnika.

Paketi informacija se identificiraju sa njihovim konekcijskim identifikatorom i sa njihovim slijednim (sekvencijskim) brojem u toku konekcije.

Mreža vodi brigu o otvaranju i zatvaranju konekcija i osigurava da se paketi prime u istom redoslijedu u kome su poslani, rekonstruirajući izgubljene pakete i upravljajući protokom.

Aplikacije vide mrežu kao pouzdanu cijev. Konekcija se zatvara nakon upotrebe.

Semantika CO-prijenosa

```
Open_Channel(Node, Task, Channel_Nr);
```

```
Send_Message (Channel_Nr, Msg1);
```

```
Send_Message (Channel_Nr, Msg2);
```

```
Close(Channel_Nr);
```

Msg1 će se primiti prije Msg2, sekvenca je sačuvana.



60/69

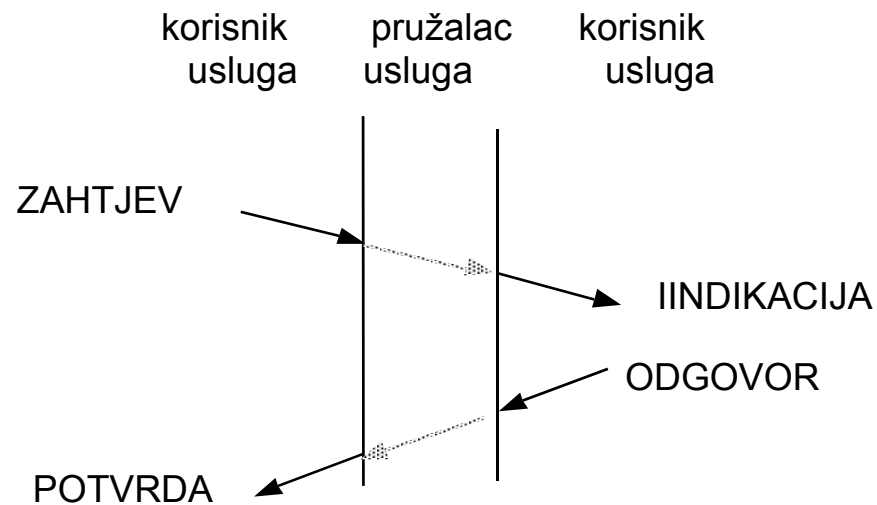
Podsloj logičkog sloja

Spojno-orijentirane usluge



Zadatak: **Upravljanje protokom i rekonstrukcija greške**

Uspostavljanje i prekidanje konekcije.
Slanje i primanje okvira.
Upravljanje protokom (Buffer control).
Ponovni pokušaj u slučaju greške.



Podsloj logičkog sloja

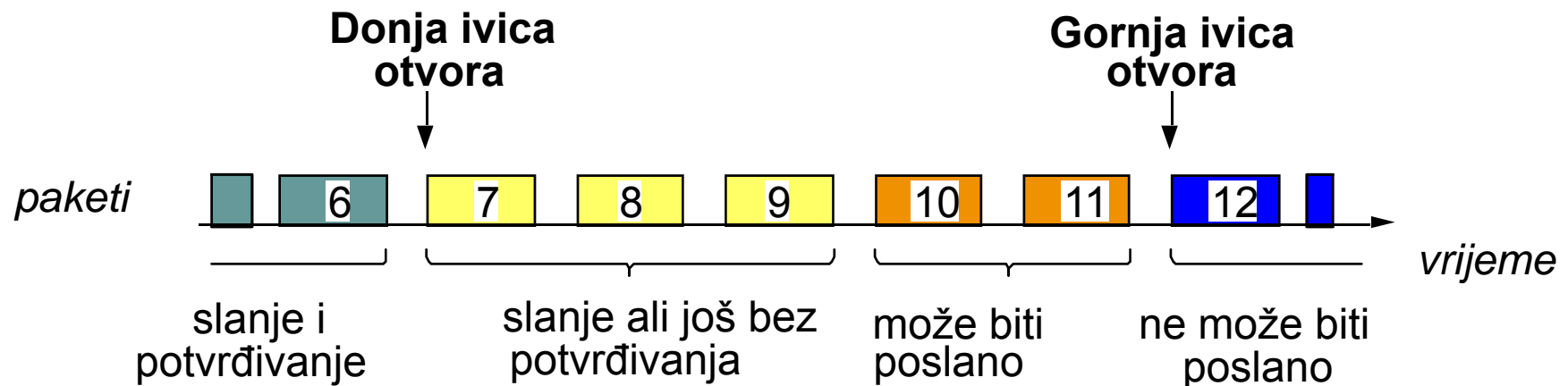
Upravljanje protokom

(= sinhronizacija na podatkovnom sloju)

“Prilagoditi brzinu predajnika brzini prijemnika”

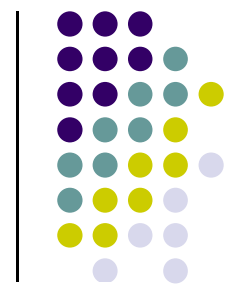
Postupci

- **Korištenje potvrde:** ne slati dok nije primljena potvrda (potvrđivanje ima dvije svrhe: rekonstrukcija greške i upravljanje protokom!)
- **Kredit:** Prijemnik indicira koliko okvira može biti prihvatljivo (protokol kliznog otvora). Pобоljšanje: promjenjiva veličina prozora.
- **Eksplicitno kočenje** (CTRL-S/ CTRL-Q)

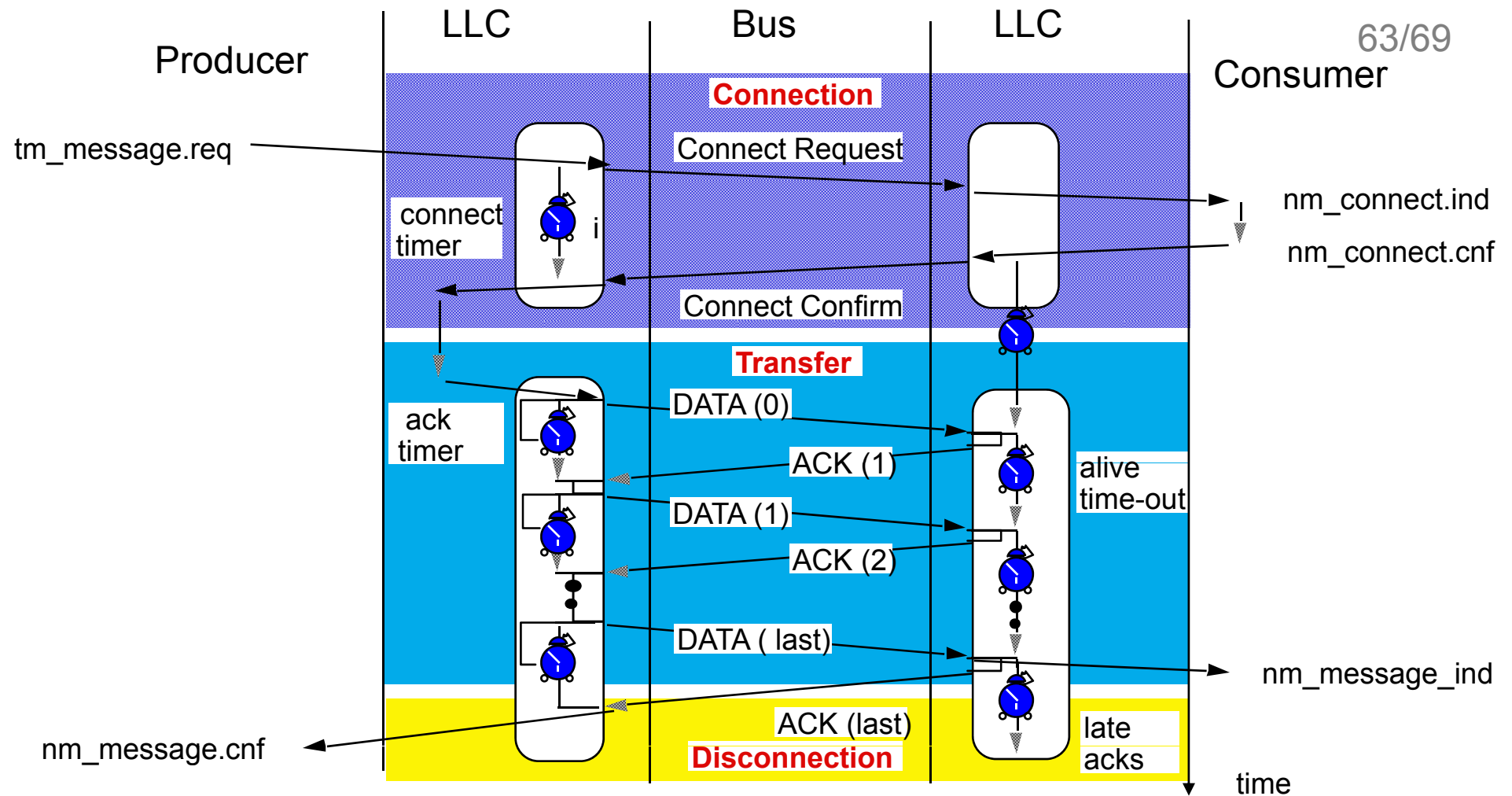


Podsloj logičkog sloja

Jednostavan prijenos sa otvorom duljine = 1



63/69



Slanje svakog paketa traje dulje od dvostrukog vremena propagacije.

Podslaj logičkog sloja

Rekonstrukcija greške

Opće pravilo: okviri pogrešnih informacija se ponavljaju
(kodovi ispravljanja greški pripadaju fizičkom sloju)



- 1) U **cikličkom prijenosu**, informacija se ne ponavlja eksplicitno u slučaju gubitka, prijemnik će primiti svježiju informaciju u sljedećem ciklusu. Upravljanje “svježinom” nadzire periode podataka u slučaju komunikacijskih prekida.
- 2) U **događajno-upravljanom prijenosu**, informacije ne mogu biti izgubljene, ponavljanje je eksplicitno:
 - a) Predajnik okvira informacija očekuje potvrdu od prijemnika, indicirajući koje informacije su primljene.
 - b) Predajnik ponavlja izgubljene informacije više puta, dok ne primi potvrdu ili dok ne istekne vrijeme.
 - c) Da bi se razlikovalo ponavljane od novih informacija, svaki paket prima slijedni broj.
 - d) Prijemnik potvrđuje ponavljanja čak i ako je on već primio ispravnu informaciju.
- 3) Kod **prijenosa u propusnom opsegu** (broadcast transmission), relativno je teško prikupiti potvrde od svih mogućih prijemnika, tako da se generalno koristi nepotvrđeni broadcast.

Od prijemnika se očekuje da protestira ako ne primi informacije.

Podsloj logičkog sloja

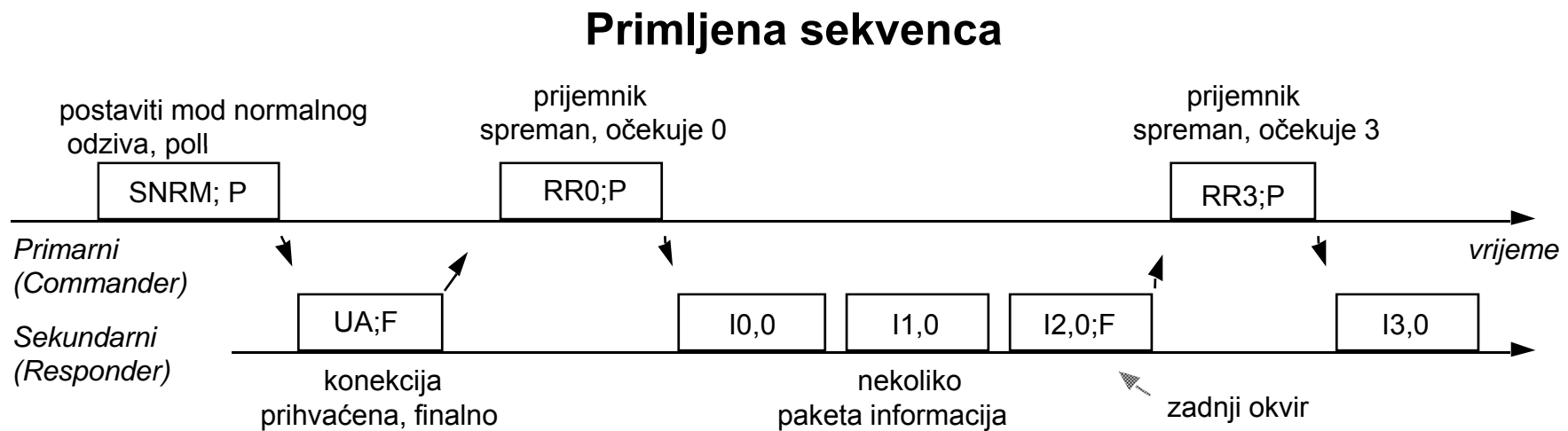
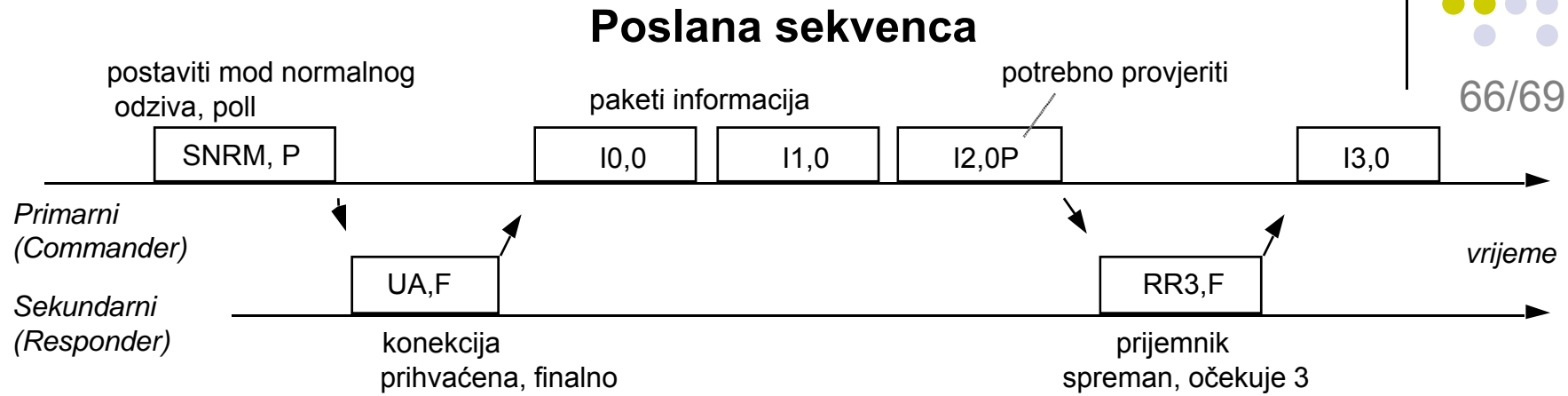
Primjer: HDLC

- HDLC (High-level Data Link Control) je izveden iz IBM-ovog SDLC (Synchronous Data Link Control) protokola.
- Ovi protokoli su razvijeni za potrebe povezivanja “point-of-sale” terminala na glavni računar.
- HDLC je najviše korišteni protokol podatkovnog sloja.
- On je osnova za CCITT-standard X25 (Telenet, Datex-P, Telepac) i koristi se u Bitnet, Appletalk, itd.
- HDLC protokol je implementiran u hardveru brojnih mikrokontrolera (npr. Zilog 80C30, Intel, Siemens 82525,... i u nekim mikroprocesorima (npr. 68360).
- HDLC je osnova za **LAN IEEE 802.2 protokol**.
- Bitovno orijentiran i koristi ubacivanje bitova.



Podslaj logičkog sloja

HDLC razmjena

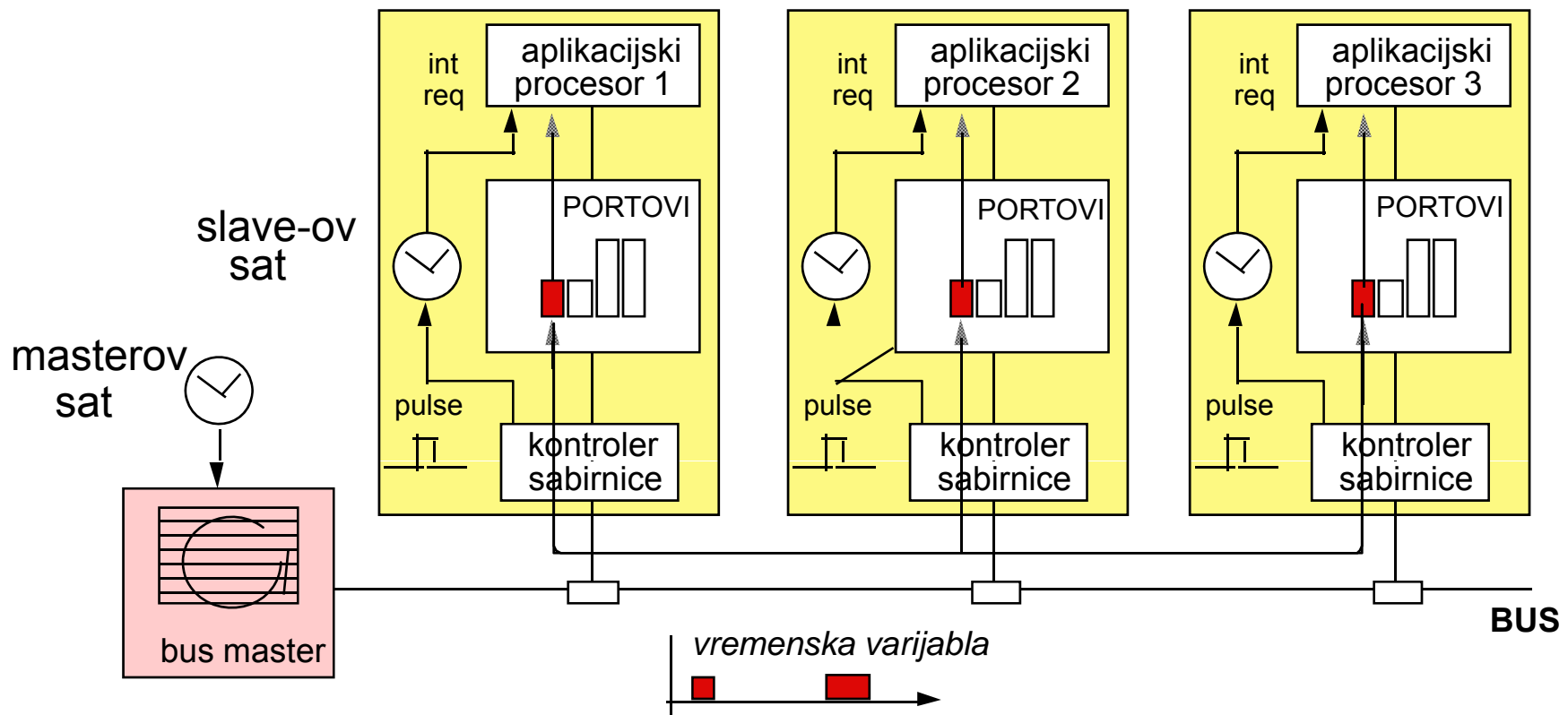


Prijenos podataka zauzima mjesto unutar "Primarnog" dijela.
 Dakle, "slanje okvira" i "primanje okvira" su podržani.

Podslaj logičkog sloja

Vremenska distribucija u jednomasterskom sistemu

U fiksnim intervalima, master emitira stvarno vrijeme kao periodičku varijablu. Kada prime ovu varijablu, kontroleri sabirnice (bus controllers) generiraju impulse koji mogu sinhronizirati sat (clock) slave-a ili generirati zahtjev za prekid.

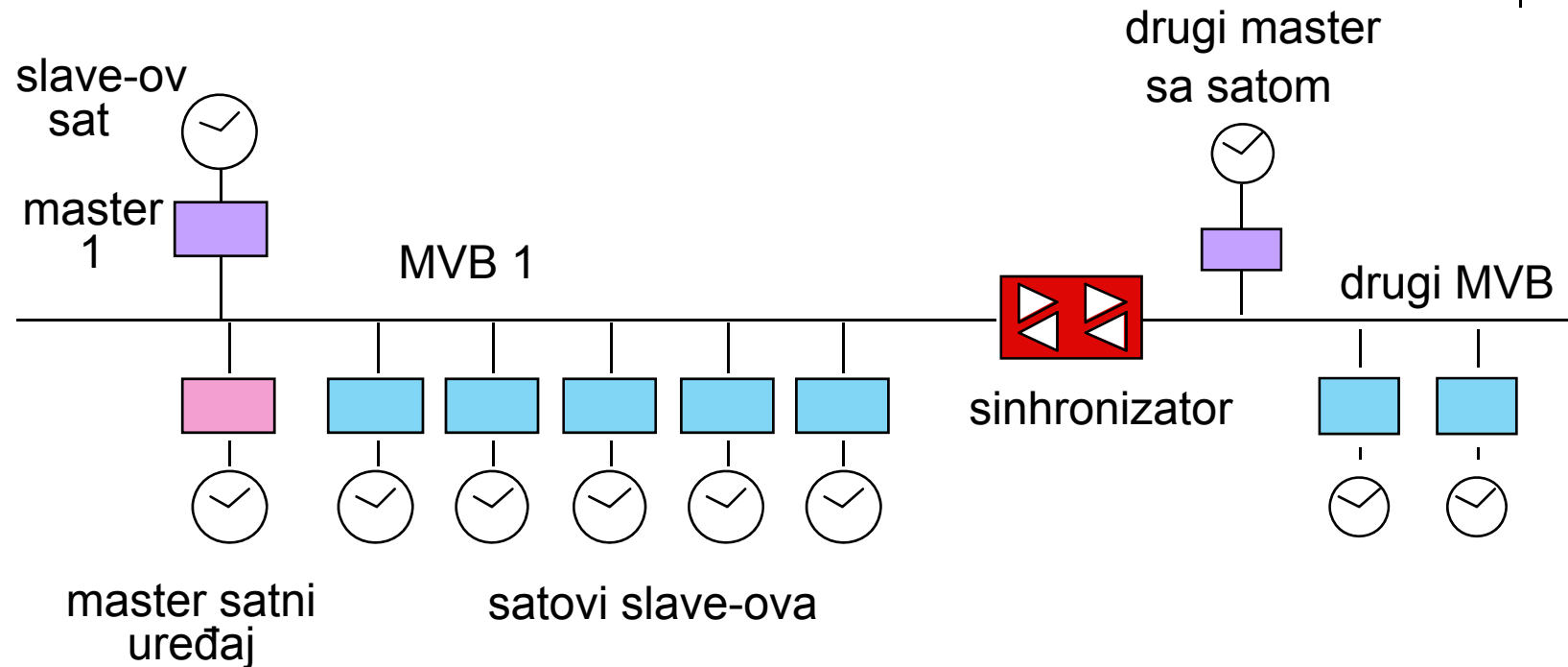


Podslaj logičkog sloja

Vremenska distribucija u jednomasterskom sistemu



68/69



Sat ne treba biti generiran Masterom, ali Master mora biti usaglašen sa satom.

Sat može sinhronizirati uzorkovanje unutar nekoliko μs kroz nekoliko sabirničkih segmenata.

Podsloj logičkog sloja

IEEE 1588 PTP sinhronizacija sata

IEEE 1588 definira PTP (Precision Time Protocol), sinhronizaciju sata koja pretpostavlja da dva uzastopna okvira imaju ista kašnjenja.

Satni uređaj (moguće vezan sa radio signalom) šalje prvi okvir, a registri u hardveru šalju informaciju o stvarnom trenutku kada je okvir poslan.

Zatim šalje drugi okvir sa tačnim vremenom u kome je poslan prvi okvir.

Premosnici i preklopnici su odgovorni za kompenziranje njihovih unutarnjih kašnjenja i šalju korigirane okvire u vremenu.

U nekim primjenama, čak i PTP protokol nije dovoljan.

U tom slučaju, sat je distribuiran (kao kod željezničke signalizacije i električkih podstanica).

Alternativno, svi uređaji primaju radio signal od GPS da bi ponovno kalibrirali njihove interne satove.

